UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

Outils logiciels temps réel pour l'assistance à la production stéréoscopique 3D

Mémoire de maîtrise
Spécialité : génie électrique

Hugo Bédard

Jury : François Michaud (directeur)
François Ferland (rapporteur)
Alain Baril (évaluateur)

Sherbrooke (Québec) Canada Juillet 2018

RÉSUMÉ

L'histoire du cinéma 3D est presque aussi longue que celle du cinéma 2D. Toutefois, ce n'est qu'avec l'utilisation des médias numériques que la commercialisation du contenu stéréoscopique 3D s'est concrétisée. Puisque la production stéréoscopique 3D nécessite l'utilisation de deux caméras, il est nécessaire de correctement ajuster celles-ci afin de produire du contenu pouvant être visualisé sans inconfort, tout en reproduisant adéquatement les caractéristiques du système visuel humain permettant la perception en profondeur. Les outils d'assistance à l'ajustement des caméras stéréoscopiques étant coûteux, la production de contenu stéréoscopique est généralement réservée aux utilisateurs expérimentés ou ayant des moyens financiers suffisants.

Afin de rendre disponible l'utilisation et l'amélioration de cette technologie, il serait pertinent de fournir des outils gratuits et libres de droit. Puisqu'il existe des bibliothèques logicielles libres pour le traitement d'images stéréoscopiques appliqué au domaine de la reconstruction 3D, ce projet cherche à évaluer la possibilité d'adapter ces algorithmes pour le développement d'outils logiciels temps réel d'assistance à la production de contenu stéréoscopique.

Pour ce faire, la détection et la correspondance de points caractéristiques sont utilisés afin de déterminer l'alignement relatif des caméras par l'estimation de la géométrie épipolaire. Les problèmes d'alignement sont par la suite corrigés par la rectification numérique des images. Afin d'obtenir une rectification stable en temps réel, les résultats montrent que des améliorations doivent être apportées aux algorithmes d'alignement et de rectification des images : 1) l'utilisation d'un détecteur de points caractéristiques alternatif non-propriétaire permettrait une meilleure performance ; 2) l'utilisation d'un algorithme alternatif pour l'estimation robuste de l'alignement des caméras permettrait une estimation sans avoir à déterminer de paramètres de façon empirique ; 3) l'utilisation d'un filtre de Kalman serait nécessaire pour une rectification stable des images lors d'une séquence vidéo. Le projet vise à l'intégration des techniques et de leurs améliorations dans une bibliothèque logicielle à code source ouvert, OpenS3D. Les fonctionnalités intégrées dans OpenS3D sont la visualisation de contenu stéréoscopique, les calculs pour l'assistance à l'alignement des caméras, la rectification numérique des images, l'analyse des profondeurs perçues pour une scène capturée. Toutes les fonctionnalités sont disponibles en temps réel à partir de l'interface utilisateur d'OpenS3D.

Puisque les améliorations apportées aux techniques de calcul d'alignement des caméras permettent d'obtenir une estimation stable et cohérente au niveau temporel, cette estimation pourrait être directement utilisée afin de corriger les erreurs d'alignement de façon automatique. Une automatisation des paramètres des caméras permettrait par exemple de filmer des scènes 3D à partir de robots mobiles. De plus, des techniques supplémentaires pourraient être ajoutées au logiciel telles que l'analyse de rivalités rétiniennes pour les différences de couleur, de luminosité ou de reflets lumineux.

Mots-clés : Cinéma 3D, Production 3D, Analyse de profondeur, Disparité, Géométrie épipolaire, Filtre de Kalman

REMERCIEMENTS

Tout d'abord, mes recherches furent possibles grâce au support financier du Fonds de recherche du Québec - Nature et technologies (FRQNT) et du Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG).

Je souhaite remercier mon directeur de maîtrise François Michaud pour sa confiance tout au long du projet ainsi que l'équipe du groupe de recherche IntRoLab pour leur soutien durant mes travaux.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ACRONYMES

Les acronymes utilisés dans ce document sont résumés ci-dessous :

Acronyme	**Définition**
3D	Trois dimensions
2D	Deux dimensions
BRIEF	*Binary Robust Independent Elementary Features*
DoG	*Difference-of-Gaussian*
GPDP	*Geometric Perceived Depth Percentage*
k-PPV	k Plus Proches Voisins
LMedS	*Least Median of Squares*
LoG	*Laplacian-of-Gaussian*
MSER	*Maximally Stable Extremal Regions*
OpenCV	*Open Computer Vision*
openMVG	*Open Multiple View Geometry*
OpenS3D	*Open Stereoscopic 3D*
ORB	*Oriented FAST and Rotated BRIEF*
RANSAC	*Random Sample Consensus*
RAM	*Random-Access Memory*
SIFT	*Scale-Invariant Feature Transform*
STAN	*Stereoscopic Analyzer*
SURF	*Speeded Up Robust Features*

CHAPITRE 1

INTRODUCTION

La perception en trois dimensions (3D) est une des capacités sensorielles les plus utiles pour analyser l'environnement qui nous entoure. La possibilité de reproduire fidèlement cette capacité sensorielle au niveau de l'affichage de contenu cinématographique a depuis longtemps été étudiée. En effet, l'histoire du cinéma 3D est presque aussi longue que celle du cinéma en 2D. Les frères Lumière ont été les premiers à montrer un film 3D à l'exposition mondiale de Paris en 1903, et le premier film complet en 3D a été produit à Los Angeles en 1922 [Zilly *et al.*, 2011a]. La technologie nécessaire pour la production et l'affichage de contenu 3D étant insuffisante à cette époque, ce n'est qu'avec la venue des médias numériques que le cinéma 3D a fait son apparition officielle de façon commerciale.

La production de films 3D à grand succès tel qu'Avatar (2009) a permis l'accroissement de l'engouement envers les médias 3D. La production de contenu stéréoscopique est toutefois plus coûteuse puisqu'elle nécessite plus de matériel (entre autres par l'utilisation de deux caméras) ainsi que des connaissances et une expérience spécialisée pour produire un contenu de qualité. À ce jour, la production 3D est principalement réservée aux compagnies américaines à fort budget : en 2015, au Québec, l'assistance des films projetés en 3D est de 88% pour les productions provenant des États-Unis, et de seulement 4% pour les productions québécoises [Observatoire de la culture et des communications du Québec, 2016]. Puisque la production cinématographique 3D est plus complexe, des recherches ont été effectuées afin d'automatiser son processus et d'optimiser le confort du spectateur en reproduisant le plus fidèlement possible l'effet de profondeur dans les images. Par contre, bien que les résultats de ces recherches soient publics, les outils nécessaires à la production cinématographique 3D restent inaccessibles ou sont disponibles de façon commerciale à coût élevé. Il est donc ardu de faire avancer la technologie ou même de l'utiliser. Par exemple, le prix du *Stereo3D Cat de Dashwood*, un logiciel d'assistance à la production stéréoscopique s'élève à $12 499 USD pour une licence complète [Dashwood Cinema Solutions, s. d.].

Il serait ainsi pertinent de fournir, de façon gratuite, les outils nécessaires à la production cinématographique 3D. Ces outils sont, par exemple, l'assistance à l'alignement des caméras stéréoscopiques, l'ajustement des couleurs et des lentilles et l'ajustement de la profondeur perçue pour assurer un confort du spectateur sur l'affichage visé. En implé-

mentant ces fonctionnalités sous forme d'outils logiciels, il serait possible de les adapter pour un utilisateur inexpérimenté qui souhaite produire du contenu 3D, pour un utilisateur professionnel plus expérimenté ou pour un contexte de recherche afin d'améliorer les technologies existantes. Ceci permettrait de rendre la technologie de cinématographie 3D accessible au même niveau que la technologie 2D l'est à ce jour.

Le but du projet de recherche est donc de produire des outils logiciels temps réel d'assistance à la production stéréoscopique 3D. Puisque les algorithmes proposés dans la littérature sont souvent incomplets pour des raisons de secret professionnel et ne sont pas adaptés pour l'analyse et la correction en temps réel du contenu stéréoscopique, il n'est pas possible d'utiliser directement ces algorithmes. Ce projet comprend donc l'adaptation de ces algorithmes pour qu'ils puissent être utilisés de façon stable dans un contexte d'analyse en temps réel du contenu stéréoscopique. Les différents algorithmes ainsi que les améliorations apportées à ceux-ci sont intégrés dans un outil unique à code source ouvert afin de rendre plus accessibles ces capacités pour la production de contenu stéréoscopique.

Le mémoire est organisé de la façon suivante. La section 2.1 décrit les concepts théoriques optiques et géométriques reliés à la perception et la production de contenu stéréoscopique. La section 2.2 analyse les solutions scientifiques et commerciales existantes afin de cibler les fonctionnalités nécessaires à la production de contenu stéréoscopique 3D, ainsi que les algorithmes nécessaires pour implémenter de telles fonctionnalités. Les outils développés ainsi que les améliorations apportées aux techniques existantes sont décrits au chapitre 3 qui présente l'article scientifique portant sur le logiciel libre réalisé.

CHAPITRE 2

OUTILS DE PRODUCTION EN CINÉMATOGRAPHIE 3D

Afin de cibler les outils permettant de faciliter la production de contenu stéréoscopique, la section 2.1 décrit le lien entre la perception en trois dimensions et la production de contenu stéréoscopique. La section 2.2 décrit ensuite les fonctionnalités des outils existants permettant d'assister à la production stéréoscopique, ainsi que les algorithmes pouvant être utilisés pour implémenter ces fonctionnalités.

2.1 Cadre de référence théorique

Afin de comprendre le lien entre la production de contenu stéréoscopique 3D et la perception de la profondeur du système visuel humain, il est nécessaire d'analyser comment le système visuel perçoit l'effet de profondeur et comment cet effet peut être reproduit sur un affichage stéréoscopique 3D.

2.1.1 Perception en trois dimensions

Telle que décrite par [Zilly *et al.*, 2011a] et [Hwang et Peli, 2014], la capacité du système visuel humain à détecter la profondeur dépend de l'interprétation de divers repères ou indices visuels reliés à la profondeur. Certains repères, soit les repères binoculaires, dépendent de la distance entre les yeux. Les repères de profondeur qui peuvent être perçus avec un oeil seulement ou à partir d'une image en deux dimensions sont des repères visuels monoculaires.

Repères visuels binoculaires

La perception de la profondeur dépend de la distance de séparation des yeux. Plus particulièrement, deux repères binoculaires dépendent de cette distance, soit la convergence et la disparité rétinienne.

- **Convergence**. La convergence est l'angle auquel les yeux doivent converger pour que l'objet d'intérêt puisse être observé comme une seule image claire. Le retour d'information des muscles reliés au mouvement fournit une information de distance.

- **Disparité rétinienne**. La distance interpupillaire cause un déplacement spatial entre les points 3D projetés sur la rétine gauche et la rétine droite. Ces déplacements se nomment disparités rétiniennes. Cette disparité est caractérisée par la différence en perspective entre l'oeil gauche et l'oeil droit due à leur distance relative.

Ces deux indices visuels sont étroitement reliés. L'objet sur lequel les yeux convergent est vu comme une seule image et ne contient aucune disparité. Les objets plus près du point de convergence possèdent une disparité (parallaxe) positive et les objets plus éloignés possèdent une disparité négative. Puisque les yeux inversent horizontalement et verticalement les images rétiniennes pour annuler l'inversement causé par la lentille oculaire, cela signifie que les objets plus éloignés sont perçus légèrement vers la droite selon la perspective de l'oeil droit comparé à l'oeil gauche. Le cerveau interprète la disparité des objets autour de l'objet sur lequel les yeux convergent pour déduire la structure 3D de la scène observée. Cette aptitude à déduire une structure 3D à partir des disparités rétiniennes se nomme stéréopsis.

Les disparités pour les objets près et éloignés relativement à la distance de fixation ont pour effet la vision double de ces objets. Cet effet est souvent imperceptible puisque les yeux s'accommodent à l'objet fixé (mise à foyer), les objets autour sont donc perçus comme étant flous. La région autour du point de convergence pour laquelle les objets peuvent être fusionnés sans image double se nomme la région fusionnelle de Panum (*Panum's Fusional Area*). De plus, les points sur l'horoptère sont perçus sans aucune disparité [Khaustova, 2015].

La figure 2.1 illustre les deux indices visuels binoculaires. Les yeux convergent sur un triangle alors que le carré est plus éloigné et le cercle est plus près de l'observateur. L'angle β correspond à l'angle de vergence nécessaire pour que les yeux puissent converger sur l'objet d'intérêt (le triangle). L'objet fixé est à l'intérieur de la région fusionnelle de Panum et sera perçu clairement tandis que les deux autres objets seront doubles et flous. La projection des objets sur les rétines montre la différence de perspective entre l'oeil gauche et l'oeil droit. La disparité rétinienne perçue d est observée par la superposition des deux images. L'objet près/éloigné est décalé à la droite/gauche (disparité positive/négative) selon la perspective de l'oeil droit par rapport à la projection sur l'oeil gauche.

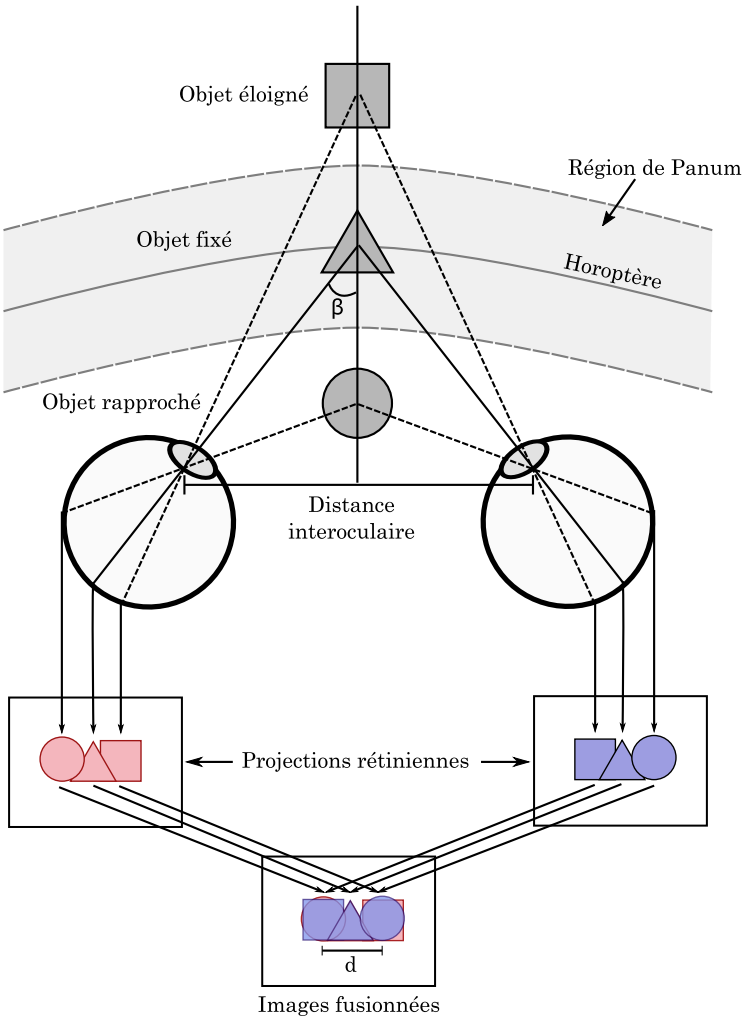

Figure 2.1 Vision binoculaire

Repères visuels monoculaires

Le système visuel humain peut aussi inférer une structure 3D à partir de différents éléments d'une seule image ou en n'utilisant qu'un oeil. Ces indices sont aussi importants et travaillent en paire avec les indices binoculaires pour l'interprétation de la profondeur [Sierra *et al.*, 2012; Zilly *et al.*, 2011a]. Ces indices peuvent être divisés en deux catégories : 1) la profondeur peut être perçue par l'observation d'images statiques avec les repères picturaux tels que l'interposition ou la perspective linéaire ; 2) les indices reliés au mouvement fournissent de l'information 3D pour des images consécutives et ont été utilisés en vision assistée par ordinateur pour estimer les caractéristiques 3D de la scène filmée [Yang *et al.*, 2012]. Certains de ces indices sont décrits plus en détail ci-dessous :

- **Accommodation**. Tel que mentionné à la section 2.1.1, l'ajustement de la distance focale de la lentille oculaire pour la mise à foyer d'objets à différentes distances est une indication de profondeur, puisque cette accommodation est directement liée à la distance.

- **Taille**. La taille d'objets observés peut aider à récupérer l'information de profondeur d'une scène. À partir de la taille des objets familiers, le système visuel utilise une connaissance antérieure de la structure tridimensionnelle afin d'estimer la taille des objets. Cette familiarité avec la taille des objets peut être utilisée pour comparer les objets connus avec des objets dont l'observateur est non familier, afin d'estimer la distance relative entre les objets à partir de leur taille relative. En comparant la taille de deux objets, l'objet le plus gros semble être le plus près. Lorsque le système visuel n'est pas familier avec l'objet observé, un gros objet sera interprété comme étant près de l'observateur.

- **Perspective**. Les lignes parallèles semblent converger avec la distance (perspective linéaire). Par exemple, lorsqu'une route est observée, les côtés de celle-ci semblent se rapprocher avec la distance et semblent converger au point de fuite. Les caractéristiques géométriques de la perspective linéaire ont été utilisées, par exemple, pour recréer l'effet de perspective en deux dimensions sur des peintures durant la renaissance italienne.

- **Luminosité et ombrages**. La directivité de la lumière et des ombres créées par les objets contiennent de l'information par rapport à la structure 3D d'une scène. Par exemple, la manipulation de la structure 3D perçue est utilisée en pratique par les maquilleurs pour exagérer les traits faciaux pour les spectateurs loin de la scène.

- **Interposition/Occultation**. Les objets devant l'observateur bloquent la lumière des objets derrière. Le système visuel s'attend donc à ce que les objets plus près cachent les objets derrière ceux-ci lorsqu'ils se chevauchent.

- **Gradient de texture**. Les détails de la texture d'une surface peuvent être perçus clairement pour les objets près de l'observateur et ne sont pas visibles pour les objets lointains. L'impression de profondeur est plus remarquée lorsqu'une même surface s'étend de près de l'observateur à une distance considérable. Par exemple, la granularité d'une route devient imperceptible avec la distance.

- **Parallaxe de mouvement**. Lorsqu'un observateur bouge relativement à des objets stationnaires, les objets plus près auront une vitesse relative plus élevée que les objets plus éloignés. Certaines techniques de vision assistée par ordinateur telles que le flux optique (*optical flow*) utilisent cette propriété afin d'évaluer les mouvements relatifs des objets observés [Horn et Schunck, 1981]. Si le mouvement de l'observateur est connu, la

parallaxe de mouvement peut permettre de retirer de l'information absolue en terme de profondeur.

- **Perspective aérienne**. Les objets semblent se fondre dans l'atmosphère lorsqu'ils sont lointains. Par exemple, l'atmosphère est visible en observant des montagnes lointaines (brouillard de distance). La couleur et le contraste relatifs peuvent donner une impression de profondeur.

2.1.2 Contenu stéréoscopique 3D

L'affichage stéréoscopique 3D utilise les indices visuels pour reproduire l'effet de profondeur à partir de deux images projetées sur un même écran avec une image visible seulement pour l'oeil gauche et l'autre image visible seulement pour l'oeil droit. La présente section explique comment la géométrie 3D peut être extraite d'une scène filmée et reproduite sur un affichage 3D.

Géométrie de la capture 3D

Afin de capturer la structure 3D d'une scène, les propriétés binoculaires des yeux sont imitées en filmant la scène avec deux caméras séparées par une distance interaxiale ou entraxe (b_c). Il existe deux configurations de caméras connues pour la capture de contenu stéréoscopique : la capture en convergence et la capture parallèle. Ces deux configurations peuvent être décrites par un modèle de caméra de type sténopé (*pinhole*) afin de dériver les relations géométriques entre les différents paramètres de capture stéréoscopique [Hartley et Zisserman, 2003]. Les deux caméras sont généralement disposées sur un rig 3D qui permet d'ajuster le positionnement relatif des caméras tel que l'alignement, la convergence et la distance interaxiale.

- **Configuration en convergence**. Telle qu'illustrée à la figure 2.2.a, la configuration en convergence utilise un angle de convergence α afin de positionner le plan de convergence Z_{0_c} à la distance où les axes des deux caméras convergent. Les points sur ce plan apparaissent sur le plan de l'écran avec une disparité nulle. Les objets à une distance de z_c sont capturés avec une disparité d. Le déplacement en profondeur cause un déplacement horizontal sur le plan de l'écran, de direction opposée entre l'oeil gauche et l'oeil droit. Par contre, la rotation des axes des caméras induit une distorsion de perspective causant des disparités verticales dans les images qui devront être supprimées avec une rectification des images, pour éviter les inconforts oculaires.

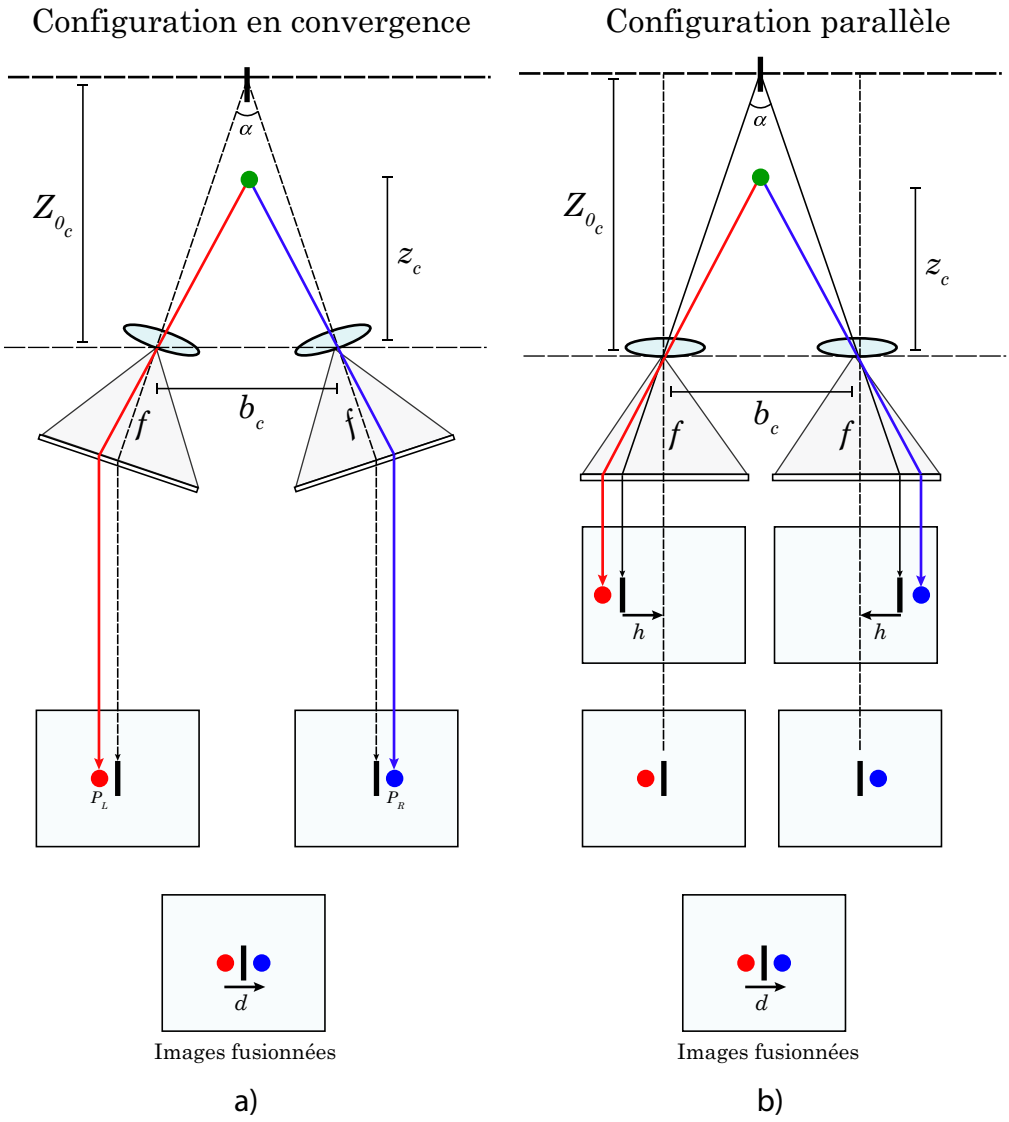

Figure 2.2 Géométrie de la capture

- **Configuration parallèle**. Contrairement à la configuration en convergence, la configuration parallèle montrée à la figure 2.2.b n'induit pas de distorsion de perspective. Pour cette raison, cette configuration est une alternative généralement préférée à la configuration en convergence puisqu'elle produit du contenu stéréoscopique de qualité supérieure [Woods *et al.*, 1993]. Le décalage du capteur (ou des pixels) de l'image peut être utilisé afin d'ajuster le plan de l'écran à la distance désirée en déplaçant de h le point focal virtuel désiré au centre de l'image. Pour une configuration parallèle, tous les points sont projetés devant le plan de l'écran pour un décalage nul ($h = 0$). Le décalage permet donc entre autres de déplacer les objets derrière le plan de l'écran.

En assumant que les paramètres sont de même unité et que les images sont rectifiées, les relations géométriques entre les paramètres de scène et de capture permettent de déterminer les disparités dans les images résultantes. La disparité d d'un point 3D distant de z_c à l'axe parallèle aux axes des caméras dépend de la distance du plan de convergence Z_{0_c}, de la distance focale de la caméra f et de la distance interaxiale des caméras b_c telle que décrit par l'équation (2.1) [Zilly *et al.*, 2011a].

$$d = b_c \cdot f \cdot (1/Z_{0_c} - 1/z_c) = 2 \cdot -b_c \cdot f/z_c \tag{2.1}$$

Géométrie de l'affichage 3D

Des relations géométriques similaires peuvent être utilisées pour calculer la profondeur perçue en fonction des paramètres d'affichage et des disparités des images. Telle qu'illustrée par la figure 2.3, la profondeur perçue z_e dépend de la distance d'observation Z_{0_e}, de la distance interoculaire b_e et de la disparité de l'écran (*parallaxe*) d_s, et peut être calculée à partir de l'équation (2.2) [Koppal *et al.*, 2011].

$$z_e = \frac{Z_{0_e} \cdot b_e}{b_e - d_s} \tag{2.2}$$

Cette équation permet de déterminer que les points avec une disparité négative apparaissent devant le plan de l'écran, tandis que les points avec une disparité positive apparaissent derrière le plan de l'écran. La disparité de l'écran est obtenue à partir de la disparité de l'image en pixels et du ratio S_r entre la largeur de l'écran w_d et la largeur du capteur de la caméra w, comme l'exprime l'équation (2.3).

$$d_s = S_r \cdot d = (w_d/w) \cdot d \tag{2.3}$$

Il est important de noter que les images captées par les caméras sont reflétées horizontalement et verticalement avant d'être projetées sur l'affichage, tel qu'illustré à la figure 2.3.a, afin d'éliminer l'inversion causée par la lentille de la caméra.

Géométrie du contenu stéréoscopique

En combinant les équations (2.1), (2.2) et (2.3), il est possible de déterminer la profondeur perçue en fonction de tous les paramètres stéréoscopiques de capture et d'affichage selon l'équation (2.4).

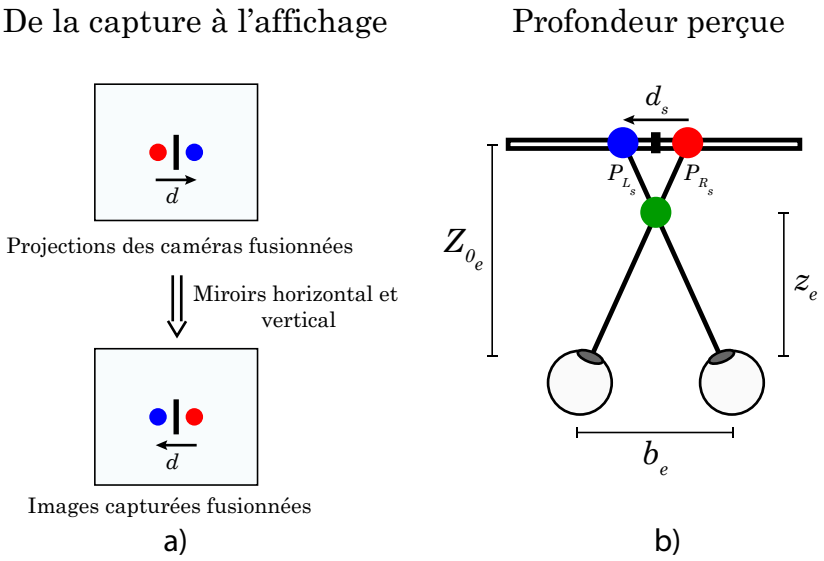

Figure 2.3 Géométrie de l'affichage 3D

$$z_e = \frac{Z_{0_e} \cdot b_e}{b_e - S_r \cdot (2 \cdot h - b_c \cdot f / z_c)} \tag{2.4}$$

Les stéréographes utilisent cette équation pour adapter la capture d'images stéréoscopiques (entraxe des caméras, décalage des images) aux différents contextes d'affichage (distance de l'observateur à l'écran et taille de l'écran) et à la scène filmée (distance des objets les plus près et les plus éloignés).

2.1.3 Règles de production stéréoscopique

Afin de reproduire correctement l'effet de profondeur à partir du contenu stéréoscopique, il est nécessaire de respecter certaines règles afin d'éviter des conflits entre les repères visuels et d'assurer une plage de profondeurs confortable pour l'observateur. Les différentes règles de production de contenu stéréoscopiques sont décrites dans cette section.

Conflits visuels

Puisque la reproduction de profondeur à partir du contenu stéréoscopique reste une illusion créée par différents repères visuels, des conflits entre les repères monoculaires et binoculaires peuvent causer une incapacité à percevoir la profondeur. Ces conflits devraient être évités dans un contexte de production stéréoscopique 3D.

- **Conflit de vergence et accomodation**. Puisque les images de droite et de gauche sont projetées sur un seul écran, les yeux doivent ajuster leur foyer sur l'écran tout en

convergeant à une distance différente pour fusionner les deux images. La distance entre le point de convergence et le point d'accomodation devrait être minimisée afin de prévenir les conflits de vergence et d'accomodation. La zone qui minimise cet effet est la zone de confort et est illustrée à la figure 2.4 [Lang *et al.*, 2010]. La profondeur perçue devrait donc rester près du plan de l'écran.

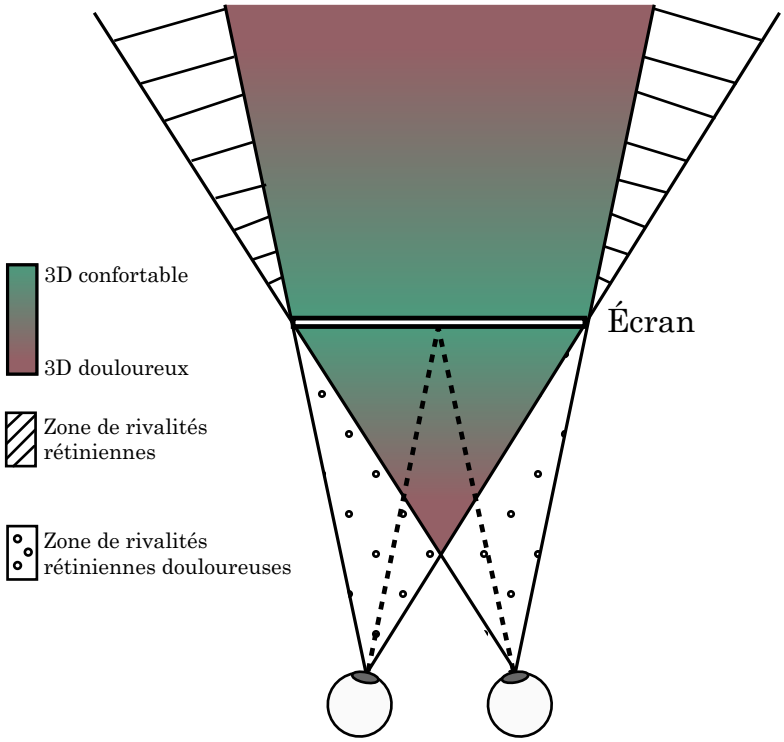

Figure 2.4 Zone de confort stéréoscopique [Lang *et al.*, 2010]

- **Non-respect des fenêtres stéréoscopiques**. Puisque l'occultation est un indice monoculaire important [Tsirlin *et al.*, 2010], les conflits entre les indices binoculaires et l'interposition monoculaire devraient être considérés lors de la production de contenu stéréoscopique. Une situation où de tels conflits se produisent est lorsqu'un objet projeté devant le plan de l'écran est partiellement coupé par les bords de l'écran. Il y a alors une violation des fenêtres stéréoscopiques : il y a un conflit entre l'indice binoculaire qui projette l'objet plus près que l'écran et l'indice monoculaire qui indique que l'écran est plus près que l'objet, puisque l'écran « cache » l'objet. De plus, les points sur les côtés de l'écran causent une rivalité rétinienne puisque cette partie de l'image n'est visible que par un oeil seulement. Ces problèmes peuvent altérer la reproduction efficace de la profondeur et devraient être évités en gardant les objets qui sont coupés par les bords de l'écran près du plan de l'écran. Il est aussi possible d'introduire une fenêtre flottante reproduite devant l'objet en utilisant une disparité appropriée [Sierra *et al.*, 2012].

Planification de la profondeur - Production stéréoscopique

Telle que décrite à la section 2.1.2, la reproduction de la profondeur par les médias stéréoscopiques dépend des paramètres de capture telle que la distance interaxiale, et des paramètres d'affichage tels que la taille de l'écran et la distance d'observation. Lors de production de contenu 3D, ces paramètres doivent être correctement ajustés, généralement en suivant les étapes suivantes :

1. Planification de la taille de l'écran maximale et de la distance d'observation.
2. Planification du budget de profondeur avec une plage de profondeur optimisant le confort de l'observateur.
3. Ajustement de la distance interaxiale pour obtenir cette plage de profondeur pour les objets les plus près et les plus éloignés.
4. Ajustement du décalage des images pour positionner le plan de l'écran (plan à parallaxe nul).

2.2 Outils d'assistance à la production stéréoscopique

Les étapes nécessaires afin d'assurer le confort du spectateur lors de la production de contenu stéréoscopique nécessitent de multiples calculs et une compréhension des concepts géométriques reliés à la reproduction de l'effet de profondeur. Certains chercheurs ont développé des modèles mathématiques et des outils pour assister les stéréographes[1] à la production de contenu stéréoscopique de meilleure qualité. Ces outils permettent une production plus rapide avec un plus petit risque d'erreur.

Les outils d'assistance à la production de contenu stéréoscopique peuvent être divisés en quatre catégories : passif, actif, acquisition de données, postproduction. Les outils de production passifs nécessitent une action manuelle de l'utilisateur afin de produire du contenu 3D de qualité, et n'agissent qu'à titre indicatif seulement. Les outils de production actifs appliquent automatiquement les corrections nécessaires aux paramètres stéréoscopiques ou aux images durant la capture. L'acquisition de données est nécessaire pour analyser les images durant la capture ou pour produire des métadonnées qui peuvent être utilisées en postproduction, et les outils de postproduction permettent de corriger les images après la capture. Les solutions existantes sont énumérées au tableau 2.1 et sont décrites plus en détail à la section 2.2.1. Ce tableau permet de rapidement identifier quelles solutions sont

[1] Stéréographe : Responsable de la profondeur lors de la production cinématographique 3D.

plus complètes au niveau des fonctionnalités, et quelles fonctionnalités sont les plus importantes par leur fréquence d'apparition plus élevée dans le tableau. Chaque fonctionnalité peut être identifiée selon la notation suivante :

Outils de production passifs (P) – Calculs stéréoscopiques et analyse de profondeur

1. Alignement des caméras (position, rotation, zoom).
2. Détection de rivalités binoculaires (couleur, distortion géométrique).
3. Calcul de paramètres stéréoscopiques (distance interaxiale des caméras, décalage des images).
4. Visualisation des paramètres stéréoscopiques (distance relative au plan de l'écran).
5. Analyse/visualisation de la profondeur des images 3D capturées.
6. Analyse/visualisation en temps réel de la profondeur d'images 3D.

Acquisition de données (D)

1. Acquisition en temps réel d'images stéréoscopiques.
2. Acquisition de métadonnées du rig 3D.

Outils de production actifs (A) – Corrections automatiques durant la capture

1. Correction/rectification en temps réel des images.
2. Ajustement automatique des paramètres stéréoscopiques durant la capture.

Outils de postproduction (P-P) – Corrections/adaptations après la capture

1. Correction/rectification d'images après la capture.
2. Adaptation des niveaux de profondeur avec cartes de profondeur et *inpainting*.
3. Adaptation des niveaux de profondeur avec déformation d'images.

Tableau 2.1 Outils de production 3D

Solution	Description	Fonctionnalités
Analyse 3D (outils commerciaux)		
[Dashwood Cinema Solutions, s. d.]	*Stereo3D Cat*	P.1, P.2, P.4, P.5, P.4, P.6
[Sony, s. d.]	*Multi Image Processor*	P.1, P.2, P.3, P.6, A.1, A.2, D.1, D.2
[3ality Technica, s. d.]-1	*Multi Image Processor*	P.1, P.2, D.1, D.2, A.1
[Binocle, s. d.]-1	*TaggerLive*	P.1, P.2, P.6, A.1
Analyse 3D		
[Zilly *et al.*, 2011b]	Analyseur stéréoscopique (points caractéristiques)	P.1, P.3, P.4, P.5, P.6, A.1
[Koppal *et al.*, 2011]	Éditeur centré sur le visualisateur	P.3, P.4, P.5, P-P.3
[Guan *et al.*, 2016]	Unité de profondeur perçue (GDPD)	P.3, P.4
Automatisation de la production		
[3ality Technica, s. d.]-2	Rigs 3D complètement motorisés	A.2, D.2
[Ilham et Chung, 2013]	Rig 3D semi-automatique	A.2 (semi-automatique)
[Heinzle *et al.*, 2011]	Système informatique de caméras stéréo	P.2, P.3, P.4, P.6, A.1, A.2
[Mielczarek *et al.*, 2015]	Processeur vidéo SDI temps réel	D.1, P.6
Postproduction		
[Kim *et al.*, 2008]	Reconstitution d'imagerie stéréoscopique	P-P.2
[Lang *et al.*, 2010]	Cartographie des disparités non linéaires	P-P.3
[Chang *et al.*, 2011]	Éditeur pour l'adaptation à l'affichage	P-P.3
Postproduction (outils commerciaux)		
[Dashwood Cinema Solutions, s. d.]	*Stereo3D Toolbox*	P.2, P.5, P-P.1, P-P.2
[Binocle, s. d.]-2	*DisparityKiller*	P-P.1

2.2.1 Outils de production stéréoscopique 3D

Afin de produire du contenu stéréoscopique de qualité, il est premièrement nécessaire d'ajuster correctement les caméras. Les disparités verticales ainsi que les défauts géométriques doivent être éliminés, et les deux caméras doivent produire des images quasi identiques au niveau de la couleur et de l'illumination (fonctionnalités P-1, P-2). Certains outils commerciaux existent afin d'assister lors de cette étape de calibration des caméras :

- Le *Stereo3D Cat* de Dashwood [Dashwood Cinema Solutions, s. d.] est un logiciel qui permet de visualiser en temps réel, sur un ordinateur, le contenu stéréoscopique (en mode anaglyphe, image de gauche/droite seulement ou côte-à-côte). Il est aussi possible de visualiser les disparités en pixels de l'image filmée, et de visualiser la position d'objets virtuels près et éloignés selon la perspective du visualisateur en fonction des mesures de la scène prises par le stéréographe. Par contre, il ne permet pas d'estimer la profondeur perçue d'objets filmés selon la perspective de l'observateur.

- Le *Multi Image Processor* de Sony [Sony, s. d.] est une solution matérielle qui permet l'acquisition des images stéréoscopiques, l'affichage des statistiques sur la profondeur des images filmées et la détection de configuration problématique. Contrairement à [Dashwood Cinema Solutions, s. d.], il peut apporter des corrections automatiques d'erreurs d'alignements physiques ou optiques aux images capturées. Il offre aussi le calcul automatique d'entraxe de caméra et l'ajustement automatique du rig 3D par la communication avec certains modèles spécifiques de moteurs.

- Le *Stereo Image Processor* de 3ality [3ality Technica, s. d.] est une solution matérielle alternative au *Multi Image Processor* de Sony. Il permet l'analyse et la correction automatique d'erreurs d'alignements physiques et optiques des caméras ainsi que la capture de métadonnées. Il permet en plus l'alignement automatique des lentilles de caméras lors d'opérations de zoom. Par contre, il ne permet pas l'analyse en profondeur ainsi que l'ajustement automatique de la distance interaxiale des caméras.

- Le *TaggerLive* de Binocle [Binocle, s. d.] permet, comme les autres solutions, de corriger les erreurs d'alignements physiques (rotation, translation) et offre une visualisation de la profondeur perçue par un histogramme de profondeur.

Certaines solutions dans la littérature tentent de fournir des outils d'assistance aux stéréographes similaires aux solutions commerciales. La solution se rapprochant le plus des outils commerciaux en termes de fonctionnalités est le *Stereoscopic Analyzer* STAN [Zilly *et al.*, 2011b]. STAN est un système d'assistance pour la production stéréoscopique qui

utilise la détection de points caractéristiques pour optimiser l'alignement des caméras en estimant la matrice fondamentale qui relie les deux caméras[2]. Ce système permet d'éliminer les disparités verticales et les distorsions géométriques avec la rectification d'images, et détecte la position d'objets près et éloignés afin de calculer automatiquement la distance interaxiale optimale des caméras. Contrairement aux solutions commerciales, les algorithmes utilisés par ce système sont décrits explicitement dans l'article. Les fonctionnalités mentionnées dans l'article sont assez similaires aux solutions commerciales telles que [Dashwood Cinema Solutions, s. d.] (voir tableau 2.1).

Puisque la perception en 3D dépend fortement des propriétés de visualisation (taille de l'écran, distance de l'observateur), certains systèmes ont été développés afin d'estimer la qualité des images stéréoscopiques pour différents contextes de visualisation :

- Koppal, en collaboration avec Microsoft Research et l'Université de Washington, a développé un éditeur de vidéos stéréoscopique qui permet de planifier ainsi que de corriger les prises de vue en visualisant l'effet 3D des images filmées selon la perspective du spectateur en fonction des paramètres de visualisation [Koppal *et al.*, 2011]. Avec quelques images clées, il est possible, à partir de l'interface utilisateur, d'observer le positionnement des différents points de l'image dans l'espace de visualisation (relatif au plan de l'écran). Il est aussi possible d'observer l'impact de changements des paramètres de caméras (distance interaxiale) sur l'effet de relief, et de corriger les scènes où les paramètres stéréoscopiques sont peu adaptés pour les conditions de visualisation visées. Contrairement aux solutions commerciales et à [Zilly *et al.*, 2011b], cette solution ne permet pas l'analyse en temps réel de la profondeur des images stéréoscopiques. L'avantage de cette méthode est qu'elle permet de visualiser la scène selon le point de vue de l'observateur.

- Une unité de mesure a aussi été établie afin de quantifier la profondeur perçue par l'observateur, soit le pourcentage géométrique de profondeur perçue ou *Geometric Perceived Depth Percentage* (GPDP) [Guan *et al.*, 2016]. Cette évaluation de la profondeur perçue dépend de la profondeur des objets filmés, de la distance focale de la caméra et des paramètres de visualisation tels que la taille de l'écran et la distance de l'observateur. L'algorithme développé est similaire à [Koppal *et al.*, 2011] malgré qu'il soit utilisé dans un contexte d'images générées par ordinateur. Puisque la profondeur des images est déjà connue lors de la génération d'images, le calcul est fortement allégé, ce qui explique pourquoi l'implémentation de cet algorithme peut être utilisé dans un contexte temps réel,

[2]La matrice fondamentale permet de trouver pour chaque pixel d'une première image, une ligne (épipolaire) sur laquelle se trouve le point de correspondance dans une seconde image [Hartley et Zisserman, 2003].

contrairement à l'algorithme de [Koppal *et al.*, 2011] qui doit préalablement déterminer la profondeur de la scène à partir des images stéréoscopiques.

2.2.2 Outils d'automatisation de production stéréoscopique 3D

Suite à l'alignement et l'ajustement des caméras ainsi qu'au calcul des paramètres stéréoscopiques adéquats (entraxe, convergence), il est possible de produire du contenu stéréoscopique de qualité. Par contre, lorsque les scènes filmées sont fortement dynamiques en profondeur, il est possible que les paramètres stéréoscopiques doivent varier durant la prise de vue. De plus, ces paramètres varient généralement entre les prises de vue puisqu'ils doivent être adaptés à la scène filmée en fonction des objets les plus près et les plus éloignés. Il est donc pertinent d'automatiser l'ajustement des paramètres stéréoscopiques en analysant en temps réel la plage de profondeur de la scène filmée.

Un paramètre stéréoscopique important est la distance interaxiale des caméras. Afin de pouvoir ajuster automatiquement ce paramètre, il est premièrement nécessaire de motoriser le rig 3D pour pouvoir varier la distance entre les caméras. Certaines compagnies produisent de tels rigs motorisés pouvant être ajustés automatiquement [3ality Technica, s. d.]. Il existe de plus quelques prototypes de semi-automatisation de rig 3D [Ilham et Chung, 2013].

Afin d'obtenir un ajustement automatique des paramètres stéréoscopiques, il est nécessaire de développer un système qui permet de fermer la boucle de contrôle en intégrant dans un même système, l'acquisition et le traitement d'images en temps réel, le calcul des propriétés stéréoscopiques de l'image et le contrôle des moteurs du rig 3D. Des chercheurs ont développé, en collaboration avec Disney Research Zurich, un même système permettant l'ajustement automatique de la distance interaxiale ainsi que la convergence des caméras [Heinzle *et al.*, 2011]. Une telle automatisation permet, entre autres, le suivi d'un acteur pour l'ajustement automatique du plan de l'écran, ce qui n'est pas possible avec un système manuel. Ce système utilise le processeur vidéo temps réel *Real-Time SDI Video Processor*, aussi développé par Disney Research Zurich [Mielczarek *et al.*, 2015], afin de faire l'acquisition en temps réel des images, de les analyser et de les rectifier pour déterminer la profondeur des objets filmés. Cet article montre que l'ajustement automatique des paramètres stéréoscopiques nécessite plusieurs sous-systèmes. Le développement de ce système plus complexe a été possible puisque Disney Research Zurich a développé en parallèle les sous-systèmes nécessaires. Ces systèmes n'ont pas été mis à la disposition d'autres chercheurs qui voudraient faire avancer le domaine de l'automatisation de la capture stéréoscopique.

Certaines solutions dans la littérature peuvent être utilisées afin de reproduire un tel système à boucle fermée. L'analyseur stéréoscopique STAN, mentionné à la section 2.2.1, inclut le calcul des paramètres 3D optimaux, ce qui a permis l'intégration de ces techniques au système d'assistance développé. Les algorithmes mentionnés par [Guan *et al.*, 2016; Koppal *et al.*, 2011] ont aussi été utilisés pour optimiser les paramètres 3D en fonction du point de vue de l'observateur.

2.2.3 Postproduction et post-traitement 3D

Malgré l'existence d'outils permettant de choisir les paramètres de caméras optimaux pour une prise de vue, il est parfois nécessaire d'adapter une production stéréoscopique 3D pour plusieurs tailles d'écran. Les plages de disparités recommandées ne sont pas les mêmes pour une représentation en cinéma que pour une écoute à la télévision. Il est donc nécessaire de pouvoir adapter les plages de disparités des images stéréoscopiques après la capture. Les outils de postproduction stéréoscopiques sont donc axés sur l'adaptation du 3D pour différentes tailles d'écran.

La première technique pour adapter le contenu 3D consiste à générer une carte de profondeur dense (*dense depth map*), de projeter les pixels des caméras en 3D, pour ensuite les reprojeter sur des plans de caméras avec différents paramètres stéréoscopiques, par exemple pour modifier la distance entre les caméras [Kim *et al.*, 2008]. Puisque certains éléments de la scène peuvent être cachés par les objets filmés, il est nécessaire avec cette méthode de peindre (*inpainting*) les éléments manquants dans la scène après la transformation.

La méthode alternative consiste à déformer les images aux endroits où la disparité des images doit être modifiée. Éviter de déformer les parties de l'image aux caractéristiques plus importantes assure que la déformation ne sera pas remarquée à l'oeil nu. Cette technique est implémentée par Lang *et al.* [Lang *et al.*, 2010] de Disney Research Zurich en déformant l'image avec l'utilisation de correspondances de caractéristiques pour modifier les disparités de l'image de façon non linéaire. Chang *et al.* [Chang *et al.*, 2011] utilisent une méthode similaire afin de modifier les disparités d'images stéréoscopiques pour les adapter à un écran ayant un format d'image différent. Bien que la technique de Chang *et al.* évite d'introduire des distorsions spatiales, elle n'est applicable qu'à des images, tandis que la méthode de Lang *et al.* est mieux adaptée pour les vidéos en assurant la cohérence de la profondeur dans le temps (continuité temporelle).

2.2.4 Algorithmes reliés à la calibration/alignement des caméras

Certains algorithmes de vision assistée par ordinateur sont pertinents afin de développer des outils d'assistance à la production cinématographique 3D. La détection de points caractéristiques permet l'alignement des caméras et l'évaluation de la profondeur perçue dans les images stéréoscopiques. Cette section décrit comment les différents algorithmes peuvent être appliqués à un contexte d'assistance à la production cinématographique 3D.

Points caractéristiques des images

Afin de détecter les erreurs d'alignements des caméras, il peut être nécessaire de détecter et de correspondre des points caractéristiques invariants localement dans les images [Zilly *et al.*, 2010], c'est-à-dire des points d'une image qui peuvent être retrouvés dans une seconde image prise d'une même scène malgré une différente perspective, une rotation, un déplacement ou la présence de bruit. Ces points caractéristiques pourraient aussi être utilisés afin de calculer les disparités des images.

Pour la détection et la description des points caractéristiques dans une image, il existe plusieurs techniques qui ont chacune leurs avantages en termes de performance et d'invariants (translation, rotation, différence d'échelle, transformation affine, bruit) [Grauman et Leibe, 2011]. Il s'agit de déterminer les invariants nécessaires pour une application donnée afin de choisir l'algorithme de détection et de caractérisation le plus approprié au niveau des performances et des invariants. Miksik et Mikolajczyk [Miksik et Mikolajczyk, 2012] offrent une comparaison des performances et de la précision des différents algorithmes. Puisqu'il existe un bon nombre de méthodes différentes, seules les méthodes principales sont décrites ci-dessous.

- **Détection des points caractéristiques.** La première étape pour déterminer les caractéristiques locales d'images est de trouver des points d'intérêts (*keypoints*) qui peuvent être retrouvés dans différents contextes de luminosité, selon différents de point de vue ou lors de présence de bruit. Les méthodes existantes sont par exemple la détection avec le détecteur Hessien (*Hessian detector*) [Beaudet, 1978] qui trouve des points ayant de fortes dérivées dans des directions orthogonales, ou le détecteur de Harris qui trouve des points se rapprochant le plus des coins dans l'image [Harris et Stephens, 1988]. Les points de Harris sont préférables lorsqu'il est nécessaire de retrouver précisément l'emplacement des coins, tandis que les points Hessiens permettent d'obtenir des points additionnels qui ne sont pas nécessairement des coins et ainsi couvrir une plus grande surface des objets.

- **Invariance aux changements d'échelle.** Il n'est par contre pas possible de précisément retrouver les points caractéristiques obtenus avec le détecteur Hessien ou le détecteur de Harris si l'échelle varie trop entre les deux images. La détection automatique de l'échelle peut être utilisée afin de caractériser un point en fonction de son échelle à partir de son voisinage. À ces fins, le détecteur *Laplacian-of-Gaussian* (LoG) peut être utilisé [Lindeberg, 1998]. Celui-ci détermine plutôt les maximums locaux de la fonction LoG dans l'espace de l'échelle. Le détecteur *Difference-of-Gaussian* (DoG) est une approximation de LoG qui cherche à maximiser la différence de la fonction gaussienne à différentes échelles [Crowley *et al.*, 2002]. Il existe aussi des détecteurs Hessiens et de Harris plus robustes par l'utilisation de l'espace de l'échelle [Mikolajczyk et Schmid, 2004].

- **Invariance aux transformations affines.** Pour des situations où il y a une différence de perspective entre deux images dont il faut comparer les points caractéristiques, il est possible d'utiliser le détecteur de Harris et le détecteur Hessien adaptés aux transformations affines [Mikolajczyk et Schmid, 2004], ou d'utiliser MSER (*Maximally Stable Extremal Regions*) qui est plus approprié pour des caméras stéréoscopiques à large distance interaxiale [Matas *et al.*, 2004].

- **Descripteurs de points caractéristiques.** Afin de décrire les points caractéristiques après leur détection pour pouvoir ensuite les comparer, il est possible d'encoder leurs caractéristiques avec des descripteurs invariants à l'échelle tels que *Scale Invariant Feature Transform* (SIFT) [Lowe, 1999] ou *Speeded-Up Robust Features* (SURF) [Bay *et al.*, 2008], qui est une implémentation plus rapide de SIFT. Par contre, ces algorithmes étant propriétaires, il est donc nécessaire d'utiliser des alternatives gratuites telles que *Binary Robust Independent Elementary Features* (BRIEF) [Calonder *et al.*, 2010] ou *Oriented FAST and Rotated BRIEF* (ORB) [Rublee *et al.*, 2011].

Rectification numérique des images

Afin de minimiser les disparités verticales occasionnées par un problème d'alignement des caméras, il est possible de rectifier de façon numérique la paire d'images en appliquant une transformation sur chacune des images. Les techniques appropriées dans un contexte de production de contenu stéréoscopique sont celles qui permettent de minimiser les distorsions des images en plus de préserver la distance du plan de l'écran, et ainsi les profondeurs reproduites. Pour ce faire, Zilly *et al.* [Zilly *et al.*, 2010] ont proposé une technique afin d'estimer l'alignement des caméras qui permet d'obtenir directement les paramètres de rectification et de corriger le désalignement restant suite à un ajustement manuel des ca-

méras. Cette technique est utilisée entre autres par le système d'assistance STAN [Zilly *et al.*, 2011b] ainsi que le système développé par Disney Research Zurich [Heinzle *et al.*, 2011]. Une technique alternative a été proposée par [Mallon et Whelan, 2005], mais ne peut être calculée directement à partir de l'alignement estimé des caméras.

Algorithmes reliés au calcul de disparité

Puisque le confort de l'utilisateur est directement relié aux disparités des images stéréoscopiques, il est nécessaire d'analyser les différentes techniques qui permettent d'obtenir ces disparités.

En utilisant les points caractéristiques calculés à l'aide d'une des méthode de la section 2.2.4, il est possible de déterminer les disparités des points caractéristiques des images en trouvant les correspondances entre les points puis en soustrayant leur valeur horizontale. Pour trouver les correspondances entre les points, il est possible d'utiliser une méthode de force brute en testant toutes les possibilités ou en utilisant une technique du plus proche voisin (k-PPV) qui est optimisée pour un grand nombre de points. Ces techniques sont coûteuses puisqu'elles doivent chercher dans l'espace 2D de l'image, mais permettent de trouver les correspondances sans une connaissance a priori de la matrice fondamentale.

Lorsque la matrice fondamentale est connue, il est possible de restreindre la recherche de points avec la contrainte épipolaire, ce qui réduit à une dimension la recherche puisque la ligne épipolaire sur laquelle le point correspondant se trouve sur la seconde image est connue [Hartley et Zisserman, 2003]. De plus, lorsqu'une rectification des images est possible, les points caractéristiques seront positionnés sur une même rangée dans les deux images, ce qui permet une recherche encore plus efficace.

Il est aussi possible de calculer une carte de disparité dense plutôt que d'utiliser des points dispersés lorsque les images sont préalablement rectifiées. Cette carte de disparité peut être calculée à partir d'algorithmes tels que *Belief Propagation* [Yang *et al.*, 2010]. Les avantages d'utiliser les cartes de disparités denses sont que celles-ci sont plus rapides à calculer puisqu'elles ne nécessitent pas de recherche de points caractéristiques et qu'elles contiennent de l'information de profondeur pour tous les pixels de l'image. Cette information est par contre bruitée puisqu'il n'est pas possible de déterminer la profondeur des pixels de l'image qui ne peuvent être identifiés par leur texture. L'utilisation de points caractéristiques a pour sa part l'avantage de ne calculer les disparités que sur les points dont une correspondance est possible, ce qui donne des résultats moins bruités en assurant une bonne précision pour la disparité des points dont la correspondance a pu être trouvée.

Les points caractéristiques peuvent aussi être utilisés afin de calculer l'alignement relatif entre les caméras.

2.2.5 Outils de vision stéréoscopique par ordinateur

En ce qui concerne le domaine plus général de la vision assistée par ordinateur, certains logiciels libres existent afin d'extraire de l'information 3D d'images stéréoscopiques. Par exemple, OpenCV et openMVG [Itseez, 2015; Moulon *et al.*, 2017] permettent de trouver des caractéristiques communes aux deux images. Ces bibliothèques logicielles implémentent de façon efficace les algorithmes de traitement d'images présents dans la littérature tels que la détection de points caractéristiques, la calibration des caméras, l'estimation de la matrice fondamentale. La structure 3D de la scène observée peut être estimée à partir des caractéristiques obtenues à l'aide de ces outils si les paramètres intrinsèques et extrinsèques des caméras sont préalablement déterminés par une calibration des caméras. Une des différences entre le domaine de la vision assistée par ordinateur et le domaine de la production de contenu stéréoscopique est le type de caméras utilisé. Les caméras spécialisées au domaine de la vision 3D sont fabriquées de façon à minimiser les erreurs d'alignement, contrairement aux médias stéréoscopiques qui nécessitent l'ajustement variable et manuel de la position relative des caméras, d'où l'utilité de développer des outils afin de détecter les erreurs d'alignement des caméras lors de la calibration manuelle de celles-ci. De plus, lors de la production de contenu stéréoscopique, le confort de l'observateur doit être pris en compte pour ajuster les paramètres stéréoscopiques, ce qui n'est pas le cas dans un contexte plus général de vision par ordinateur, par exemple en robotique où il n'y a pas de contexte de visionnement du contenu stéréoscopique.

2.2.6 Accessibilité aux outils présentés

Les solutions mentionnées dans cette section montrent les avancements dans le domaine de l'assistance à la production stéréoscopique 3D. Ces solutions proviennent en partie de compagnies, en partie d'articles d'organismes publics et d'articles affiliés à des organismes privés tels que Disney. Comme le montre le tableau 2.2, bien que certains articles expliquent les algorithmes utilisés pour implémenter les fonctionnalités visées, la plupart des implémentations sont soit omises par secret professionnel, soit laissées comme exercice au lecteur. Puisque certaines fonctionnalités de base telles que la calibration de caméras sont nécessaires afin de pouvoir développer des fonctionnalités plus complexes comme l'automatisation des paramètres stéréoscopiques, cela peut rendre difficile l'avancement de la technologie dans ce domaine.

Tableau 2.2 Analyse des outils de production 3D

Solution	Accessibilité	Généralisabilité
Analyse 3D (outils commerciaux)		
[Dashwood Cinema Solutions, s. d.]	Algorithmes privés	Oui - Si les algorithmes étaient publics
[Sony, s. d.]	Algorithmes privés	Non - Spécifique au matériel propriétaire
[3ality Technica, s. d.]	Algorithmes privés	Non - Spécifique au matériel propriétaire
[Binocle, s. d.]	Algorithmes privés	Oui - Si les algorithmes étaient publics
Analyse 3D		
[Zilly *et al.*, 2011b]	Implémentation privée	Oui - Équations mathématiques/algorithmes
[Koppal *et al.*, 2011]	Implémentation privée	Oui - Équations mathématiques/algorithmes
[Guan *et al.*, 2016]	Implémentation privée	Oui - Équations mathématiques/algorithmes
Automatisation de la production		
[3ality Technica, s. d.]	Matériel propriétaire	Non - Solution matérielle seulement
[Ilham et Chung, 2013]	Nécessite du matériel propriétaire	Oui - Applicable à du matériel autre
[Heinzle *et al.*, 2011]	Implémentation privée	Non trivialement - Spécifique au matériel
[Mielczarek *et al.*, 2015]	Nécessite du matériel propriétaire	Non - Matériel seulement
Postproduction		
[Kim *et al.*, 2008]	Implémentation privée	Oui - Équations mathématiques/algorithmes
[Lang *et al.*, 2010]	Implémentation privée	Oui - Équations mathématiques/algorithmes
[Chang *et al.*, 2011]	Implémentation privée	Oui - Équations mathématiques/algorithmes
Postproduction (outils commerciaux)		
[Dashwood Cinema Solutions, s. d.]	Algorithmes privés	Oui - Si les algorithmes étaient publics
[Binocle, s. d.]	Algorithmes privés	Oui - Si les algorithmes étaient publics

2.3 Conclusions

Selon la revue de littérature présentée, les problèmes soulevés sont les suivants :

- **Accessibilité et généralisation**. Il n'existe présentement aucun outil ou bibliothèque logicielle indépendante du matériel utilisé (modèle de caméra, matériel d'acquisition d'images) permettant l'assistance à la production stéréoscopique 3D dont le code source est disponible publiquement. La plupart des solutions sont disponibles sous forme de produits commerciaux. Le code source est généralement privé et dépend de matériel propriétaire. Cela signifie que pour innover dans le domaine, il est nécessaire de faire l'achat de solutions propriétaires de type boîte noire, ou de réimplémenter les algorithmes présents dans la littérature pour pouvoir produire du contenu stéréoscopique 3D de qualité suffisante.
- **Les outils de calibration sont essentiels à la production 3D**. La perception 3D est reproduite par une illusion du système visuel humain et certains repères visuels doivent être respectés pour recréer l'effet de profondeur. La qualité de production affectant directement le confort de l'utilisateur, un standard minimal de qualité doit donc être respecté pour prévenir l'inconfort du spectateur (vision double, nausées, étourdissements, maux oculaires).
- **Recoupements avec les outils de vision assistée par ordinateur**. Il existe plusieurs solutions logicielles pour la vision stéréoscopique assistée par ordinateur appliquées au domaine de la robotique et de la reconstruction 3D [Itseez, 2015; Moulon *et al.*, 2017]. Il n'existe par contre aucune solution équivalente appliquée à la production stéréoscopique 3D. Les notions mathématiques et algorithmiques de ces deux domaines étant très similaires, certains algorithmes pourraient être réutilisés pour produire les fonctionnalités appliquées au domaine de la production 3D.

CHAPITRE 3

IMPLÉMENTATION DES TECHNIQUES ET AMÉLIORATIONS

3.1 Avant-propos

Auteurs et affiliations :

Hugo Bédard : étudiant à la maîtrise, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique.

Jean-Samuel Lauzon : étudiant à la maîtrise, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique.

François Michaud : professeur, Université de Sherbrooke, Faculté de génie, Département de génie électrique et de génie informatique

Date de soumission :

23 janvier 2018

Revue :

ACM Transactions On Graphics (TOG)

Titre en anglais :

OpenS3D, an Open-Source Real-Time Assistance Framework for Stereoscopic Content Production

Contribution au document :

Afin de pallier les limitations présentées à la section 2.3, des outils à code source ouvert d'assistance en temps réel à la production stéréoscopique 3D sont développés. Les outils développés ainsi que les améliorations qui ont été apportées afin que ces outils respectent les contraintes de code ouvert et de stabilité d'analyse en temps réel sont présentées dans cet article. Ceux-ci permettent de s'assurer que les repères visuels binoculaires sont respectés par l'alignement adéquat des caméras, la rectification des problèmes d'alignement restants et l'ajustement de la plage de disparités afin de maintenir la profondeur reproduite dans

la zone de confort stéréoscopique. Ainsi, leur utilisation rend possible l'obtention d'un contenu stéréoscopique de qualité optimale directement lors de la prise de vue afin de minimiser les ajustements en postproduction. L'article montre l'impact des améliorations apportées sur la qualité de l'estimation de l'alignement et des paramètres de rectification. L'implémentation des outils développés est décrite et sa performance est mesurée.

Résumé français :

L'estimation de la géométrie épipolaire modélisée de façon articulaire est une technique utilisée pour l'alignement de caméras stéréoscopiques. Malgré que cette problématique peut sembler réglée, le cinéma 3D étant bien établi depuis plusieurs années, l'instabilité numérique de cette technique demeure problématique au niveau de la stabilité de la rectification numérique des images appliquée en temps réel et à notre connaissance, aucune solution n'a été apportée dans la littérature. L'implémentation existante de cette technique est privée, ce qui rend difficile son amélioration et la création de nouveaux outils pour l'assistance à la production stéréoscopique. Pour résoudre ces problèmes, nous présentons une librairie logicielle à code source ouvert, OpenS3D, qui inclut deux techniques améliorées : 1) l'implémentation de l'estimation continue de l'articulation représentant la géométrie épipolaire et des paramètres de rectification avec l'aide d'un filtre de Kalman, et 2) l'implémentation temps réel de l'analyse des profondeurs selon le point de vue de l'observateur.

3.2 Abstract

Joint estimation of the epipolar geometry is a common technique used for the alignment of a stereo camera pair for stereoscopic content production. Even if this problem could be considered as solved with 3D cinema being around for many years now, numerical instability remains problematic for real-time frame-to-frame rectification and to our knowledge has not been publicly solved in the literature. Current implementations of this technique are private, which makes it difficult to use, improve and test new assistance tools and techniques for stereoscopic content production. To address these issues, we present an open-source framework, OpenS3D, which includes two improved techniques : 1) implementation of frame-to-frame joint estimation of epipolar geometry and rectification parameters using Kalman filtering, and 2) real-time implementation of viewer-centric depth analysis.

3.3 Introduction

Stereoscopic content gained in popularity in movie theaters over the last few years. Producing such content requires technical skills and experience : camera pairs must be correctly aligned and the reproduced depth range must be adjusted to produce a comfortable viewing experience [Meesters *et al.*, 2004; Zilly *et al.*, 2011a]. For scenes with a dynamic depth range or dynamic zoom levels, 3D content must be analyzed in real-time to prevent discomfort throughout the scene.

Systems have been designed to assist stereographers with the adjustment of stereoscopic parameters. STAN [Zilly *et al.*, 2011b] is a system that analyzes and monitors stereoscopic sequences through the recovery of joint epipolar geometry from feature pairs. Recovered camera geometry can be used to digitally rectify remaining misalignments. The Closed-Loop Camera System [Heinzle *et al.*, 2011, 2016] is a stereo camera system that captures, analyzes and controls physical parameters of a stereo sequence. The Viewer-Centric Editor [Koppal *et al.*, 2011] is used offline to visualize reproduced depth for different viewing parameters such as viewer distance and screen size, to adapt stereoscopic content to the viewing context. An online version of the algorithm would be better suited for real-time analysis of stereoscopic content.

Both STAN and the Closed-Loop Camera System use the technique proposed by Zilly *et al.* [Zilly *et al.*, 2010] for joint estimation of the epipolar geometry and the rectification parameters. This technique aims to assist stereographers during the manual camera alignment process, and to digitally rectify the remaining misalignments near the rectified state.

However, this algorithm as described in the paper presents numerical instability issues which are problematic for robust estimation. In the context of real-time analysis of a video sequence, which requires frame-to-frame estimation, this issue leads to sporadic jumps and outliers for the estimated rectification parameters. Since results for only one image pair are shown in the original work, issues applying this technique to image sequences are not apparent. Even if work could be considered done for this technique with 3D cinema being around for many years now, to our knowledge, this issue has not been publicly addressed and have been mostly implemented in private commercial products. We show that without the proposed method, this important technique for 3D content production is just not viable.

As previously stated, there is to our knowledge no public implementation of Zilly *et al.*'s algorithm for joint estimation of the epipolar geometry and the rectification parameters. Since adequate camera alignment is essential to produce a comfortable viewing experience, developing assistance tools for stereoscopic content production requires implementing this technique from scratch to then work on its improvement. To address this problem, this paper presents an open-source assistance framework for stereoscopic content production, named OpenS3D [Bédard, 2018]. OpenS3D bridges the gap between Zilly *et al.*'s technique and real-time use of this technique by satisfying robustness and real-time stability requirements for quality 3D content production. Improvements are provided by the use of a Kalman filter. The OpenS3D framework also includes a real-time implementation of a viewer-centric disparity mapping algorithm for intuitive online analysis of reproduced depth.

This paper is organized as follows. Section 3.4 describes the Kalman-based algorithm for real-time joint estimation of the epipolar geometry and the rectification parameters. Section 3.5 explains the real-time viewer-centric depth analysis algorithm using disparity mapping of feature pairs. Section 3.6 presents the features implemented in the OpenS3D framework. Section 3.7 presents results using OpenS3D for camera alignment and depth analysis during stereoscopic content production.

3.4 Real-Time Estimation of Epipolar Joint Geometry and Rectification Parameters

Since stereoscopic 3D relies on horizontal displacement to reproduce depth, precise camera alignment is essential to minimize vertical disparities and to produce comfortable stereoscopic content. Manually adjusting the cameras near the rectified state before the

shot prevents from having to remove vertical disparities during post-production. However, due to mechanical constraints, it may not be possible to perfectly align the cameras : remaining misalignments can be corrected using digital rectification of the image pairs.

Manual camera alignment is time consuming and requires technical skills and experience to produce high-quality content. Using such expertise, Zilly *et al.* present a technique to assist during the camera alignment process by using joint estimation of the epipolar geometry and rectification parameters. The estimated geometry can be used to monitor which alignment parameters to adjust and to correct remaining misalignments using rectification. This technique could be done through a real-time user interface. However, as previously stated, numerical instability of this technique is problematic for use in a frame-to-frame scenario and no results for such scenario are provided in the original work. In fact, for STAN, the camera alignment is done mainly by visually inspecting the anaglyph image pairs to manually minimize vertical disparities, as stated in [Zilly *et al.*, 2011b]. The joint estimation of the epipolar geometry is only an alternative advice method to align the cameras and is usually combined with visual inspections. This requirement limits the use of this technique for automatic alignment of camera pairs.

Therefore, to address this problem using an open-source framework, OpenS3D implements the following techniques, which are explained in the following subsections :

1. Camera alignement via robust epipolar geometry estimation using :

    - Oriented FAST and Rotated BRIEF (ORB) [Rublee *et al.*, 2011], an open-source efficient alternative to Scale Invariant Feature Transform (SIFT) [Lowe, 1999] or Speeded-Up Robust Features (SURF) [Bay *et al.*, 2008].

    - Least median of squares (LMedS) [Rousseeuw et Leroy, 1987] as an alternative to random sample consensus (RANSAC) [Fischler et Bolles, 1981] for robust estimation of the epipolar geometry, as suggested by Torr *et al.* [Torr et Murray, 1997] when feature point variance is unknown.

2. A Kalman filter [Kalman *et al.*, 1960] for temporal filtering of estimated geometry to improve stability in the estimated parameters and rectification.

3. Centered rectification of image pairs using camera alignment estimated with centered point correspondences.

3.4.1 Camera Alignment via Robust Epipolar Geometry Estimation

The camera alignment estimation technique from Zilly *et al.* using joint estimation of the epipolar geometry and the rectification parameters is based on the linear approximation of the joint parameters. To minimize the effect of outlying point correspondences, this linear approximation must be estimated using a robust estimator.

Linear Approximation of Joint Parameters

In the case of stereoscopic cameras, this technique assumes that the camera pair is near the rectified state. The small angle approximation can then be used to approximate the rotation matrix. The epipolar geometry is approximated using a Taylor expansion of the fundamental matrix **F** around the rectified state. A non-zero interaxial distance is also assumed.

The joint estimation of **F** is derived from these constraints as given by (3.1) [Zilly *et al.*, 2010], parametrized in relation to joint angles roll (α_x), pitch (α_y), yaw (α_z), the translated center (c_y, c_z), the focal length ratio (r_f) and the focal distance (f). Translation and rotation parameters are shown in Figure 3.1.

$$\mathbf{F} = \begin{bmatrix} 0 & \frac{-c_z+\alpha_y}{f} & c_y + \alpha_z \\ \frac{c_z}{f} & \frac{-\alpha_x}{f} & -1 + r_f \\ -c_y & 1 & -f\alpha_x \end{bmatrix} \tag{3.1}$$

Using a set of homogeneous image points $\{\mathbf{m}_i\}, i = 1, ...n$ in the first image where $\mathbf{m}_i = \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T$ transformed to the set $\{\mathbf{m}'_i\}$ in the second image by a rotation and a non-zero translation, the linear set of equations expressed in (3.3) is derived from the epipolar

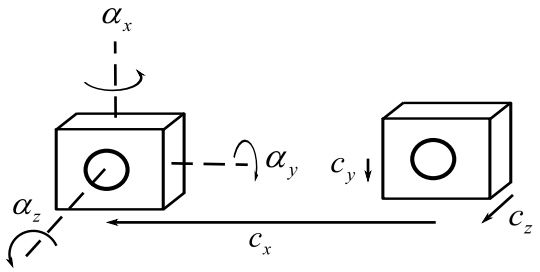

Figure 3.1 Joint parameters

constraint given by (3.2). This makes it possible to find the joint parameters by minimizing vertical disparities for $\mathbf{F}$ in (3.1).

$$\mathbf{m}_i'^T \mathbf{F} \mathbf{m}_i = 0 \tag{3.2}$$

$$v' - v = c_y \Delta u + \alpha_z u' + r_f v' - f\alpha_x + \alpha_y \frac{u'v}{f} - \alpha_x \frac{vv'}{f} + c_z \frac{uv' - u'v}{f} \tag{3.3}$$

The set of parameters $\mathbf{x}$ in (3.4) is determined by solving (3.3) using linear least-squares optimization to minimize vertical disparities $(v' - v)$.

$$\mathbf{x} = \begin{bmatrix} c_y & \alpha_z & r_f & f\alpha_x & \alpha_y/f & \alpha_x/f & c_z/f \end{bmatrix}^T \tag{3.4}$$

Point correspondences $\mathbf{m}_i \rightarrow \mathbf{m}_i'$ are found and putatively matched using a robust feature detector : it should produce as few outliers as possible and a good amount of well distributed features to obtain optimal results. Zilly *et al.* [Zilly *et al.*, 2010] recommend using SIFT combined with Difference of Gaussian interest point detection or Up-Right-SURF and the Hessian Box-Filter detector. Because SIFT and SURF are proprietary and computationally expensive feature detectors, ORB is used as a more efficient and open-source alternative [Rublee *et al.*, 2011].

Robust Estimation of Joint Parameters

Even sophisticated feature detectors produce a certain amount of outliers. To obtain a good estimation of the epipolar geometry, these outliers must be filtered out using a robust estimation algorithm [Hartley et Zisserman, 2003]. Random sample consensus (RANSAC) is typically used since it is able to cope with a large proportion of outliers. RANSAC attempts to find, with a high probability, a model that satisfies the largest number of inliers using random minimal samples. The downsides of RANSAC is that a threshold must be chosen empirically by estimating the variance.

As an alternative, LMedS [Rousseeuw et Leroy, 1987] is recommended by [Torr et Murray, 1997]. Compared to RANSAC, LMedS works also well with a large portion of outliers (under 50%) and does not require a priori estimation of variance since it uses the median as an estimate. If the proportion of outliers is over 50%, a better putative matching algorithm could be employed to reduce the number of outliers. However, in a situation where a proportion smaller than 50% cannot be ensured, RANSAC could still be a viable option

since it has been proven to work for as much as 90% of outliers, as long as the threshold is meticulously chosen. The distance metric used to evaluate inliers is the Sampson distance e (3.5) [Sampson, 1982], as recommended by [Hartley et Zisserman, 2003; Torr et Murray, 1997; Zilly *et al.*, 2010] to provide a first-order approximation of the geometric error.

$$e = \frac{(\mathbf{m}_i'^T \mathbf{F} \mathbf{m}_i)^2}{(\mathbf{F}\mathbf{m}_i)_1^2 + (\mathbf{F}\mathbf{m}_i)_2^2 + (\mathbf{F}^T\mathbf{m}_i')_1^2 + (\mathbf{F}^T\mathbf{m}_i')_2^2} \tag{3.5}$$

As stated in [Zilly *et al.*, 2010], the linear set of equation (3.3) is numerically unstable if all parameters are included since some parameters depend on all four coordinates. For example, c_z depends on both the vertical and horizontal coordinates of the point correspondences (u, u', v, v'). The focal length also depends on two tilt coefficients which makes the estimation unstable when the tilt angle is zero. This can be problematic to provide robust estimation because this algorithm compares different solutions to pick the most appropriate model. When joint parameters are not needed, it is possible to estimate $\mathbf{F}$ using the traditional 8-point algorithm [Hartley et Zisserman, 2003] which uses the epipolar constraint (3.2) to directly find the matrix elements of $\mathbf{F}$. Normalization is a necessary step to improve numerical stability for this algorithm [Hartley, 1997]. On the other hand, joint estimation of the epipolar geometry cannot use this normalization step without loosing physical meaning over the joint parametrization of $\mathbf{F}$, which explains the numerical instability ; the statistical distribution of point coordinates in (3.3) varies between feature sets.

3.4.2 Temporal Filtering of Estimated Geometry

After robust joint estimation of the epipolar geometry, vertical disparity is minimized for a specific frame but no temporal consistency is maintained. Because the optimized linear set of equations (3.3) remains numerically unstable, the estimated parameters for each specific frame may vary greatly between consecutive frames, causing a noticeable difference in orientation of the image pairs. This can be distracting to the viewer [see complementary video]. The original solution to such numerical instabilities suggested by Zilly *et al.* is to ignore and omit unstable parameters from (3.3) until it becomes stable.

The Computational Stereo Camera System [Heinzle *et al.*, 2011] uses a temporal median filter to remove outliers and a low-pass filter to remove high frequencies for the control of interaxial distance and convergence plane. A Kalman filter was considered as an alternative

filtering technique but was ruled out due to sporadic high outliers and non-linearity in the model equations.

However, for the problem of joint estimation of the epipolar geometry which is a linear process, Kalman filtering can be used as opposed to removing unstable parameters for increased numerical stability. For this linear process, a Kalman filter works well with a dynamic range of frequencies as opposed to a low-pass filter. Kalman filtering can also be used to achieve a smooth transition between rectification parameters for consecutive frames. To better explain the process, after filtering outlying feature pairs through robust estimation of epipolar geometry, it is assumed that only Gaussian noise remains in the inlier points, which propagates through the linear least-squares optimization of the joint parameters. Since the estimated parameters vary in time, the knowledge of the previous estimated state can be used to prevent sporadic jumps and outlying joint parameters estimation. This satisfies the Markovian assumption of the hidden state value $\mathbf{x}$. For the camera alignment problem, a simplified Kalman filter is derived.

Kalman filtering [Kalman *et al.*, 1960] is a two step estimation process. The first step is the Prediction step, which models the transition between each state as given by (3.6). The prediction for the current state $\mathbf{x}_k$ is derived using the state-transition model $\mathbf{G}$ applied to the previous state $\mathbf{x}_{k-1}$ plus the added process noise $\mathbf{w}_k$, which is assumed Gaussian with zero mean and covariance matrix $\mathbf{Q}$. A control-input $\mathbf{u}_k$ mapped to the state by matrix $\mathbf{B}$ is also added to the current state prediction. For camera alignment, we choose $\mathbf{u}_k$ to be directly mapped to the state space with an identity matrix ($\mathbf{I}_7$). Using $\mathbf{u}_k$ helps for convergence when the rate of change on the parameters is known, e.g., for motor control of the camera alignment parameters. Camera alignment is currently a manual process but could be made automatic by directly controlling motors using the estimated alignment error and $\mathbf{u}_k$ representing the motor movements.

$$\mathbf{x}_k = \mathbf{G}\mathbf{x}_{k-1} + \mathbf{w}_k + \mathbf{B}\mathbf{u}_k \tag{3.6}$$

Assuming constant joint parameters, $\mathbf{G}$ is $\mathbf{I}_7$ because the transition between states is negligible with assumed Gaussian noise. Covariance $\mathbf{Q}$ can be adjusted with knowledge of the parameter variation through time, i.e., for initial camera alignment adjustments.

Using the Kalman prediction hypothesis (3.6), Kalman filter equations for the Prediction step can be derived as a predicted state estimate $\hat{\mathbf{x}}_{k|k-1}$ using (3.7) and a predicted covariance estimate $\mathbf{P}_{k|k-1}$ from (3.8).

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{G}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_k = \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{u}_k \tag{3.7}$$

$$\mathbf{P}_{k|k-1} = \mathbf{G}\mathbf{P}_{\mathbf{k-1|k-1}}\mathbf{G}^T + \mathbf{Q} = \mathbf{P}_{\mathbf{k-1|k-1}} + \mathbf{Q} \tag{3.8}$$

The hypothesis for the update step is that the observation $\mathbf{z}_k$ (the estimated camera alignment, which is the output of the robust joint estimation of the epipolar geometry) is derived from the observation model $\mathbf{D}$ and the true hidden state $\mathbf{x}_k$, to which Gaussian noise $\mathbf{v}_k$ (with zero mean and covariance matrix $\mathbf{N}$) is added, as given by (3.9).

$$\mathbf{z}_k = \mathbf{D}\mathbf{x}_k + \mathbf{v}_k \tag{3.9}$$

From (3.9) are derived the equations of the second step of Kalman filtering, i.e., the Update step : innovation (3.10) and its covariance (3.11) ; the optimal Kalman gain (3.12) ; the updated state estimate (3.13) (i.e., the filtered alignment) and its covariance (3.14). Since observation and hidden state share the same form, the measurement model $\mathbf{D}$ is $\mathbf{I}_7$.

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{D}\hat{\mathbf{x}}_{k|k-1} = \mathbf{z}_k - \hat{\mathbf{x}}_{k|k-1} \tag{3.10}$$

$$\mathbf{S}_k = \mathbf{D}\mathbf{P}_{k|k-1}\mathbf{D}^T + \mathbf{N} = \mathbf{P}_{k|k-1} + \mathbf{N} \tag{3.11}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{D}^T\mathbf{S}_k^{-1} = \mathbf{P}_{k|k-1}\mathbf{S}_k^{-1} \tag{3.12}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k \tag{3.13}$$

$$\mathbf{P}_{k|k} = (\mathbf{I}_7 - \mathbf{K}_k\mathbf{D})\mathbf{P}_{k|k-1} = (\mathbf{I}_7 - \mathbf{K}_k)\mathbf{P}_{k|k-1} \tag{3.14}$$

Using (3.7-3.8,3.10-3.14), the true stable joint parameter state can be recursively estimated from one iteration at each frame, which produces a filtered smooth correction compared to the numerically unfiltered unstable joint estimation of the epipolar geometry.

3.4.3 Real-time Centered Rectification from Estimated Alignment

Once the cameras are mechanically aligned as close to the rectified state as possible, there may remain visually noticeable misalignments. The Close-Loop Camera System [Heinzle *et al.*, 2011] uses [Mallon et Whelan, 2005] and [Zilly *et al.*, 2010] to digitally correct these remaining misalignments. Because rectification parameters can be derived directly from the estimated joint geometry and produce comparable results to other rectification techniques [Hartley, 1999; Loop et Zhang, 1999; Mallon et Whelan, 2005] in term of visual distortion, as well as conserving the convergence plane [Zilly *et al.*, 2010], we chose to implement this approach in OpenS3D.

$$\mathbf{H} = \begin{bmatrix} 1 & c_y & 0 \\ -c_y & 1 & 0 \\ -c_z/f & 0 & 1 \end{bmatrix} \tag{3.15}$$

$$\mathbf{H}' = \begin{bmatrix} 1 - r_f & \alpha_z + c_y & 0 \\ -(\alpha_z + c_y) & 1 - r_f & f\alpha_x \\ \frac{\alpha_y - c_z}{f} & -\frac{\alpha_x}{f} & 1 \end{bmatrix} \tag{3.16}$$

The rectification matrices $\mathbf{H}, \mathbf{H}'$ derived from the estimated joint geometry are given by (3.15,3.16). If the joint parameters are computed from point correspondences in the image reference frame, the rectification will produce a rotation around the top-left corner of the image. Our implementation uses a centered version instead to achieve minimal changes between consecutive frames : transitions between rotations from one frame to another are less noticeable since the image rectification movements are equally balanced from the center of the image. For an image with width w and height h, by computing $\mathbf{x}$ using point correspondences with the image center $\begin{bmatrix} w/2 & h/2 \end{bmatrix}^T$ translated to the origin, it is possible to rectify the image so that the rotation is done around the center. All point correspondences are transformed with a translation matrix $\mathbf{T}$ as given by (3.17), before estimating the centered set of parameters $\mathbf{x_c}$ with (3.3) and LMedS.

$$\mathbf{m}_{i_c} = \mathbf{m}_i \cdot \mathbf{T} = \mathbf{m}_i \cdot \begin{bmatrix} 1 & 0 & -w/2 \\ 0 & 1 & -h/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

The centered fundamental matrix $\mathbf{F}_c$ and centered rectification matrices $\mathbf{H}_c, \mathbf{H}'_c$ are computed from the estimated $\mathbf{x_c}$ and can be transformed back into the image coordinate reference frame using $\mathbf{T}$, as expressed in (3.18) and (3.19).

$$\mathbf{F} = \mathbf{T}' \cdot \mathbf{F}_c \cdot \mathbf{T} \tag{3.18}$$

$$\mathbf{H} = \mathbf{T}^{-1} \cdot \mathbf{H}_c \cdot \mathbf{T} \tag{3.19}$$

3.5 Viewer-Centric Depth Analysis

After eliminating outlying point correspondences, it is possible to use the remaining inliers to analyze and monitor disparity to keep it in a comfortable range. As illustrated by Figure 3.2, depth range must be adjusted for a specific scene and a specific viewing context because reproduced depth depends on the distance of the objects in the scene, the interaxial distance and convergence of the cameras, and also on viewing parameters such as screen width w_d and viewer distance Z_e. Using these parameters as well as the interocular eye distance b_e, it is possible to map the screen disparity ($d = u' - u$) to the reproduced depth z_e. Having the ability to monitor disparity range in real-time prevents having to modify stereoscopic content in post-production and can help provide a comfortable viewing experience throughout a scene with a dynamic depth range.

The disparity mapping equation from the Viewer-Centric Editor [Koppal *et al.*, 2011] for 3D movies was modified to map image disparities in pixels to a distance in meters from the screen plane using (3.20,3.21), where S_r is the ratio between the display width w_d in meters and the camera sensor width w in pixels. These equations are used to visualize the disparity of each feature pair. The desired screen width can be set to adjust the mapping from disparity to reproduced depth provides a more intuitive visualization tool to understand the influence of camera parameters and viewing parameters on the resulting reproduced depth. In OpenS3D, this technique is implemented by computing point correspondences and filtering outliers using a robust estimation done in real-time on frames captured by the video acquisition card.

$$z_e = Z_e - \frac{Z_e b_e}{b_e - S_r d} \tag{3.20}$$

$$S_r = w_d/w \tag{3.21}$$

3.6 Implementation of OpenS3D

Algorithm 1 summarizes the use of the techniques described in Section 3.4 and Section 3.5 as implemented in a public MATLAB implementation and an open-source modular and reusable modern C++ library, referred to as the OpenS3D framework [Bédard, 2018]. OpenS3D is cross-platform (MacOS, Windows, Linux). Currently available feature detection algorithms in OpenS3D are ORB and SURF. For real-time camera alignment and rectification, users can can choose between RANSAC and LMedS.

Algorithm 1: Stereoscopic Image Sequence Analysis

Data: Roughly matched feature pairs
Result: Camera Alignment, Rectified Images, Reproduced Depth

```
while Image pair available do
    Find putative feature matches using ORB;
    Translate the origin of feature points to the image center using (3.17);
    Estimate camera alignment z_k and eliminate outliers with LMedS using (3.3) and
     Sampson distance;
    Find filtered alignment x̂_k with the next Kalman filtering iteration;
    Compute F_c from camera alignment using (3.1);
    Compute H_c and H_c' from camera alignment using (3.15,3.16);
    Translate F_c back to image coordinates F using (3.18);
    Translate H_c and H'_c back to image coordinates H and H' using (3.19);
    Rectify images around center using H and H';
    Compute viewer-centric depth from matched features inliers and viewer context using
     (3.20) ;
end
```

OpenS3D also comes with an online visualization interface to be used as a tool for real-time analysis of stereoscopic content. Figure 3.3 presents OpenS3D's visualization interface. The first feature it provides is visualization of stereoscopic content in anaglyph, side-by-side or above-below format. Horizontal image translation is also possible to translate the disparity range when filming with a parallel camera setup. Many video formats and containers are supported for offline analysis through the use of FFmpeg[1], a cross-platform, open-source audio and video library. Online analysis and monitoring of stereoscopic content are implemented using DeckLink SDK for support of Blackmagic Design DeckLink video capture

[1]FFmpeg (http ://www.ffmpeg.org)

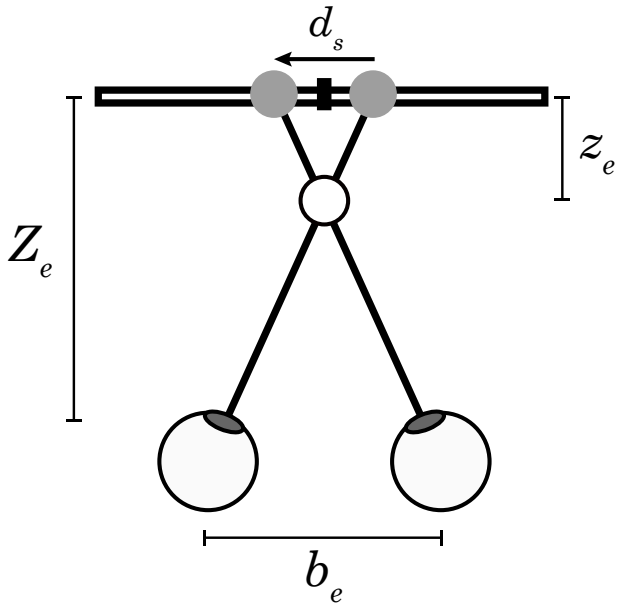

Figure 3.2 Geometry of 3D display

cards, because a 4K Extreme Capture & Playback Card was used during development. However, the modular architecture of the library would allow for additional capture interface implementations. Feature detection and putative matching are implemented using the computer vision open-source framework OpenCV [Itseez, 2015]. Kalman filter observation noise training can be done using the user interface to obtain the appropriate smoothing for specific setups and scene analysis. This training consists of measuring the covariance matrix $\mathbf{N}$ from the unfiltered joint estimation of the epipolar geometry for a specified number of frames.

The disparity range minimum and maximum values are displayed to the user (with the color range in the bottom left) and each feature can be displayed over the images with a color corresponding with their distance from the screen plane. Depending on viewing parameters, the expected disparity range can be adjusted to display a meaningful color to intuitively identify which feature points are outside the comfortable range. Figure 3.4 shows the viewer-centric window from which point correspondence disparities can be projected and visualized in meters based on (3.20). Parameters such as screen size, viewer-distance and viewer interocular distance can be modified to adjust depth to the viewing parameters.

The proposed implementation was successfully tested on OS X using a mid-2014 MacBook Pro with a 2.6 GHz Intel Core i5 processor and 8 GB of Random-Access Memory (RAM). It was also tested on Windows and Linux with a 4.0 GHz i7-6700k processor, a NVIDIA GTX-1080 video card and 32 GB of RAM. Table 3.1 summarizes the computer and ca-

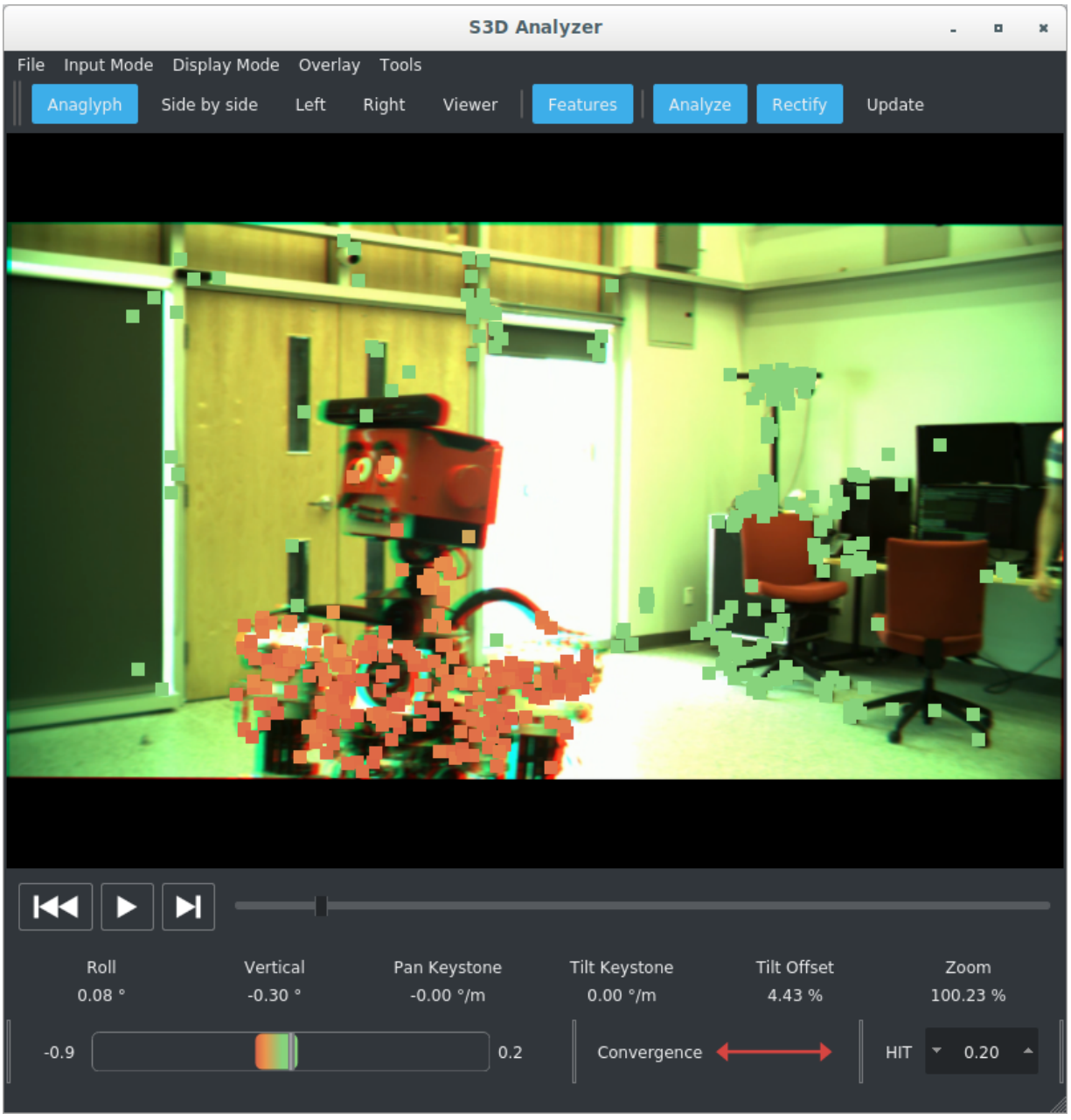

Figure 3.3 OpenS3D visualization interface

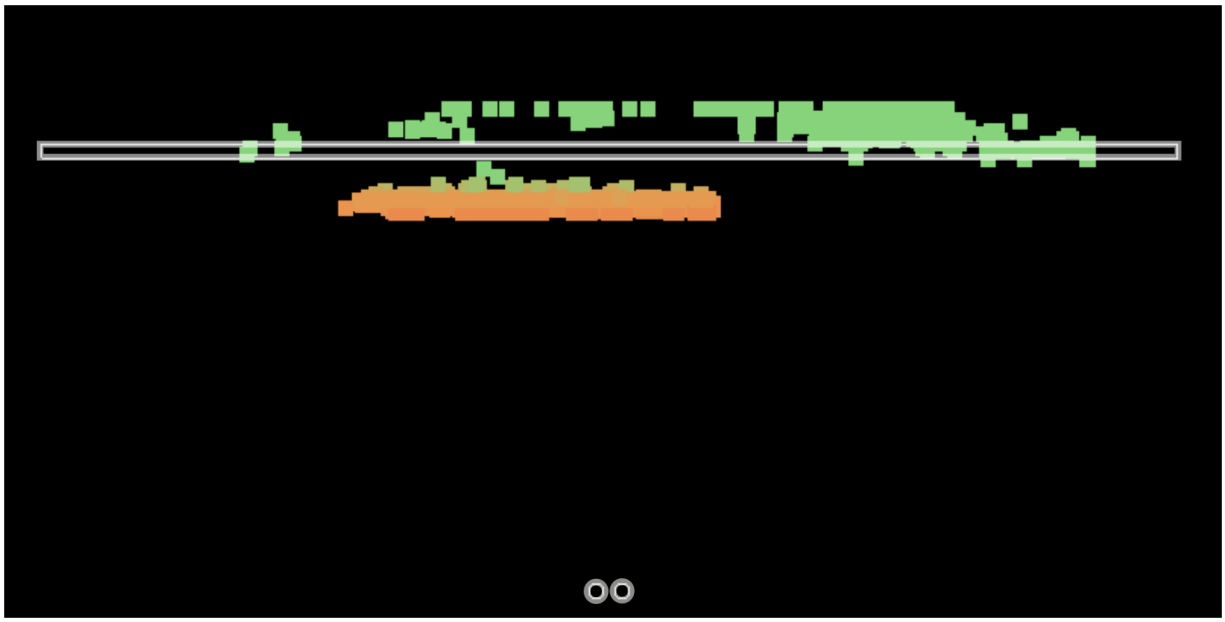

Figure 3.4 Viewer-centric window

mera equipment used. Performance of the implementation depends on multiple factors : a larger resolution, a large number of frames per second, a large number of features or a large maximum number of robust estimation iterations all contribute to the resulting computational complexity of the implementation. With this in mind, these parameters have been made customizable in OpenS3D to meet real-time requirements for different computing platforms : image size can be scaled down, the maximum number of iterations for the robust estimation can be adjusted, the maximum number of features can be set and the implementation makes it possible to skip frames when computation time exceeds frame rate period to keep a correct time scale.

3.7 Experiments

Torr *et al.*'s comparative tests for the assessment of robust epipolar geometry estimation techniques [Torr et Murray, 1997] are used to assess the accuracy and stability of the proposed technique. These tests can be used directly since the metric to compare the different techniques should measure : 1) how well these techniques estimate the epipolar geometry as compared to the real epipolar geometry and 2) how stable this estimation is.

As recommended by Torr *et al.*, point correspondences are generated using synthetic cameras similar to cameras used to capture real imagery. Table 3.2 summarizes the synthetic camera parameters used, which yield camera intrinsic parameters matrix $\mathbf{C}$ (3.22) as in [Torr et Murray, 1997].

Tableau 3.1 Equipment

Equipment	Description
3D Rig	P+S Technik freestyle 3D Rig
Cameras	Kinefinity KineMINI 4K
Camera Lenses	RED 25.0 mm
Video Capture Card	Blackmagic DeckLink 4k Extreme
Desktop Computer	
Operating System	Windows, Linux
Processor	4.0 GHz Intel Core i7-6700K
Graphics Card	NVIDIA GTX-1080
RAM	32 GB
Laptop Computer	
Operating System	OS X
Model	Mid-2014 MacBook Pro
Processor	2.6 GHz Intel Core i5
Graphics Card	Intel Iris Pro Graphics
RAM	8 GB

Tableau 3.2 Synthetic Camera Parameters

Parameter	Value
Focal Length (f)	703 pixels
Field of View	40 deg
Image Size	512×512

$$\mathbf{C} = \begin{bmatrix} 1.00 & 0.00 & 0.36 \\ 0.00 & 1.50 & 0.36 \\ 0.00 & 0.00 & 0.0014 \end{bmatrix} \tag{3.22}$$

World points $\mathbf{X}_i$ are randomly generated in $\mathbb{R}^3$ so as to be visible to both synthetic cameras. Points are projected to corresponding left and right image plane using (3.23) and (3.24) with the right camera being transformed by rotation $\mathbf{R}$ and translation $\mathbf{t}$ relative to the left camera.

$$\mathbf{m}_i = \mathbf{C} \left[\mathbf{I}|\mathbf{0}\right] \mathbf{X}_i \tag{3.23}$$

$$\mathbf{m}'_i = \mathbf{C} \left[\mathbf{R}|\mathbf{t}\right] \mathbf{X}_i \tag{3.24}$$

Gaussian noise with a variance of 2 pixels is added to point correspondences $\mathbf{m}_i, \mathbf{m}'_i$. Outliers are generated to be in random direction and between the minimum and the maximum allowable disparity range of their corresponding position in the left image. The experiments were conducted using 40% outliers, which is a large proportion of outliers, to test the robustness of the algorithm. Synthetic video sequences are generated using 200 point correspondences per frame over a total of 400 frames, and the stability of the estimated parameters is assessed as well as the Sampson distance measure for the estimated epipolar geometry. The Sampson distance is measured from the distance from the noise free synthetic points to the estimated epipolar geometry to quantify how well the estimated epipolar geometry matches the real solution generated with the synthetic cameras.

3.7.1 Synthetic Video Sequence for Constant Rectified State

In these test conditions, improvements in terms of stability and epipolar geometry estimation are assessed using a camera pair in the rectified state with a 0.5 m interaxial distance $\mathbf{t}_x$. The relative camera position is constant throughout the synthetic video sequence. The same feature points are used for the unfiltered and the filtered test cases.

For this test, the Kalman filter parameters where determined as follows. Since the covariance of the process noise between states is not modeled in our case, all terms of matrix $\mathbf{Q}$ are 0. Observation noise covariance matrix $\mathbf{N}$ was trained on a different set of a 100 frame synthetic sequence in the rectified state, which could be compared in real imagery as a

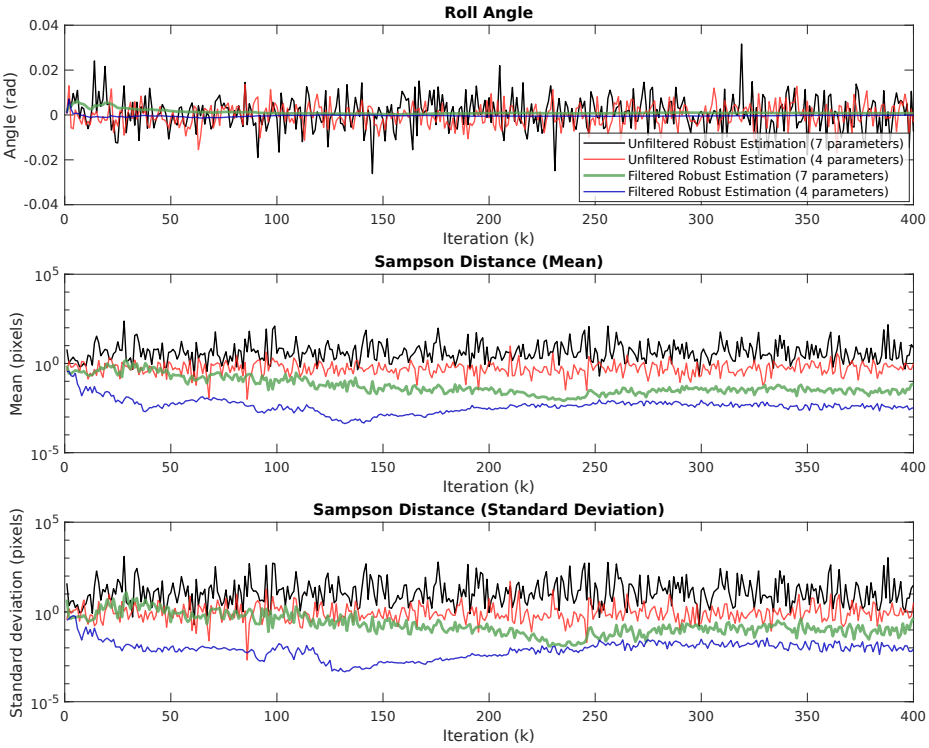

Figure 3.5 Results for the rectified state test conditions

Tableau 3.3 Metrics for the rectified state test conditions

	σ_{α_x}	$\bar{\bar{e}}$	$\sigma_{\bar{e}}$
Unfiltered (7 parameters)	$7.34 \cdot 10^{-3}$	9.89	21.34
Unfiltered (4 parameters)	$4.80 \cdot 10^{-3}$	0.70	0.71
Filtered (7 parameters)	$9.78 \cdot 10^{-4}$	0.11	0.19
Filtered (4 parameters)	$4.86 \cdot 10^{-4}$	0.0088	0.03

Tableau 3.4 Gain for the rectified state test conditions

	σ_{α_x}	$\bar{\bar{e}}$	$\sigma_{\bar{e}}$
Unfiltered (7 parameters)	-	-	-
Unfiltered (4 parameters)	1×	13×	29×
Filtered (7 parameters)	7×	86×	111×
Filtered (4 parameters)	15×	1118×	668×

video sequence of different physical scenes using the same camera setup. The observation noise measured during training was fed directly to the Kalman filter for both conducted tests and can be reused for different camera setups as shown in the second experiment (Section 3.7.2). The joint epipolar geometry was robustly estimated using LMedS.

Figure 3.5 presents the unfiltered estimation of the roll angle compared with three different methods for removing unstability : 1) by removing the three most unstable parameters (tilt keystone, pan keystone and z-parallax deformation) as suggested by Zilly *et al.*, 2) by using the proposed Kalman filter with all seven parameters, as defined in Section 3.4.2 and 3) by combining the proposed Kalman filter and the omission of the three most unstable parameters. The Sampson distance distribution (mean and standard deviation) is also shown for each frame to assess the epipolar geometry estimation error and is displayed with a logarithmic scale.

Both the filtered angle estimated with seven parameters and the filtered angle estimated with four parameters converge rapidly around the correct value under 50 iterations compared to both unfiltered robust estimations. As shown in Table 3.4, removing unstable parameters is not enough to improve stability of the estimated roll angle. Using a Kalman filter produces far superior results in terms of stability and epipolar geometry error. Compared to only the omission of parameters, the combination of the omission of parameters and the Kalman filtering methods provides the best results with a 15 times gain in terms of roll angle stability (σ_{α_x}), a 86 times improvement of the mean Sampson error ($\bar{\bar{e}}$) and a 23 times smaller Sampson error standard deviation ($\sigma_{\bar{e}}$).

3.7.2 Synthetic Video Sequence for Varying Roll Angle

For these test conditions, roll angle varies from 0 to 0.1 rad at constant speed for the first third of the 400 frames, and is left constant at 0.1 rad for the remaining frames. These test conditions aim to assess the quality of the filtered estimation with and without $\mathbf{u}_k$ compared to the unfiltered robust estimation. Since the observation noise covariance matrix was trained on the rectified state, these test conditions also validate that this training is still valid for camera geometry around the rectified state : the same matrix $\mathbf{N}$ trained for the first test conditions (Section 3.7.1) is reused for these test conditions.

To simulate motor control of the roll angle parameter, the control-input $\mathbf{u}_k$ is set to be the $\Delta\alpha_z$ used to generate synthetic point correspondences. Gaussian noise with zero mean and variance $1 \cdot 10^{-4}$ is added to this control-input to reflect an approximate knowledge of the control-input.

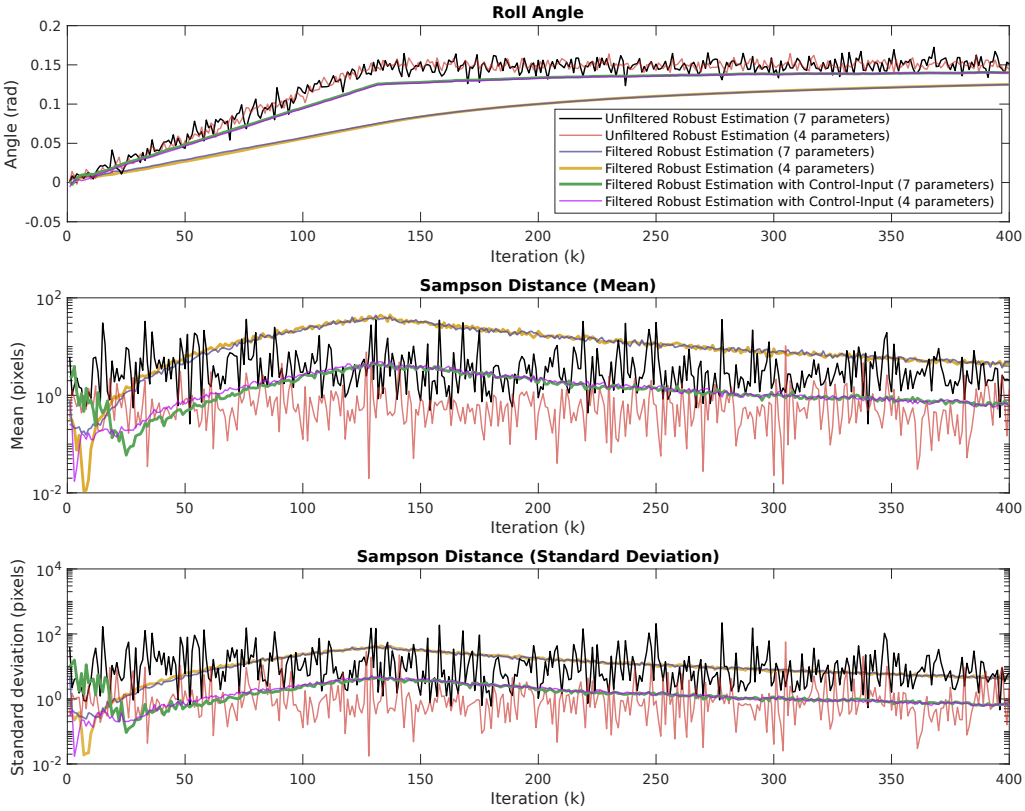

Figure 3.6 Results for the varying roll angle test condition

Figure 3.6 shows that using a Kalman filter yields the most stable roll angle and Sampson error. However, a delay is required during rapid changes to converge to the correct value. The control parameter helps reduce this convergence delay as shown by the Sampson error being smaller during the whole sequence with control input. As opposed to the rectified state test case, the difference between the Sampson error for the estimation with seven parameters and the estimation with four parameters is much less significant during camera alignment adjustments. The results for this test case show that estimating the alignment with the proposed Kalman filter with four of the seven parameters is the most efficient technique to provide a stable image sequence after camera rectification even with varying parameters. Computational complexity for the robust estimation with four parameters is lower than with all seven parameters and provides accurate and stable epipolar geometry estimation.

3.7.3 Real-Time Implementation

Figure 3.7 shows a typical example of near rectified camera setup after manual adjustments done using the provided framework in an online session. With a maximum number of features of 1000, ORB as the feature detector and a scale of 50% on an image with a resolution of 720p, it is possible to analyze and rectify the image pair as well as display the analysis results and images at a rate of approximatively 30 frames per second. This experiment was conducted using a desktop computer described in Table 3.1. The frame rate was estimated using GLXOSD, a benchmarking tool for Linux. Figure 3.7 also displays these benchmark results.

3.8 Conclusion

OpenS3D is implemented as an open source real-time assistance tool for stereoscopic content production, that provides camera alignment, rectification and viewer-centric depth analysis. To demonstrate its use, this paper presents improvements over existing techniques for real-time joint estimation of epipolar geometry and rectification parameters for the analysis of stereoscopic content. The proposed techniques aim to bridge the gap between the original work and its usability with real-time video sequences. To do so, ORB is used as an efficient and open-source alternative feature detector, and LMedS as an alternative robust estimation algorithm to RANSAC. The novel use of a Kalman filter to estimate the joint geometry from the four most stable parameters demonstrates a far more accurate frame-to-frame estimation of the epipolar geometry than by only omitting the three

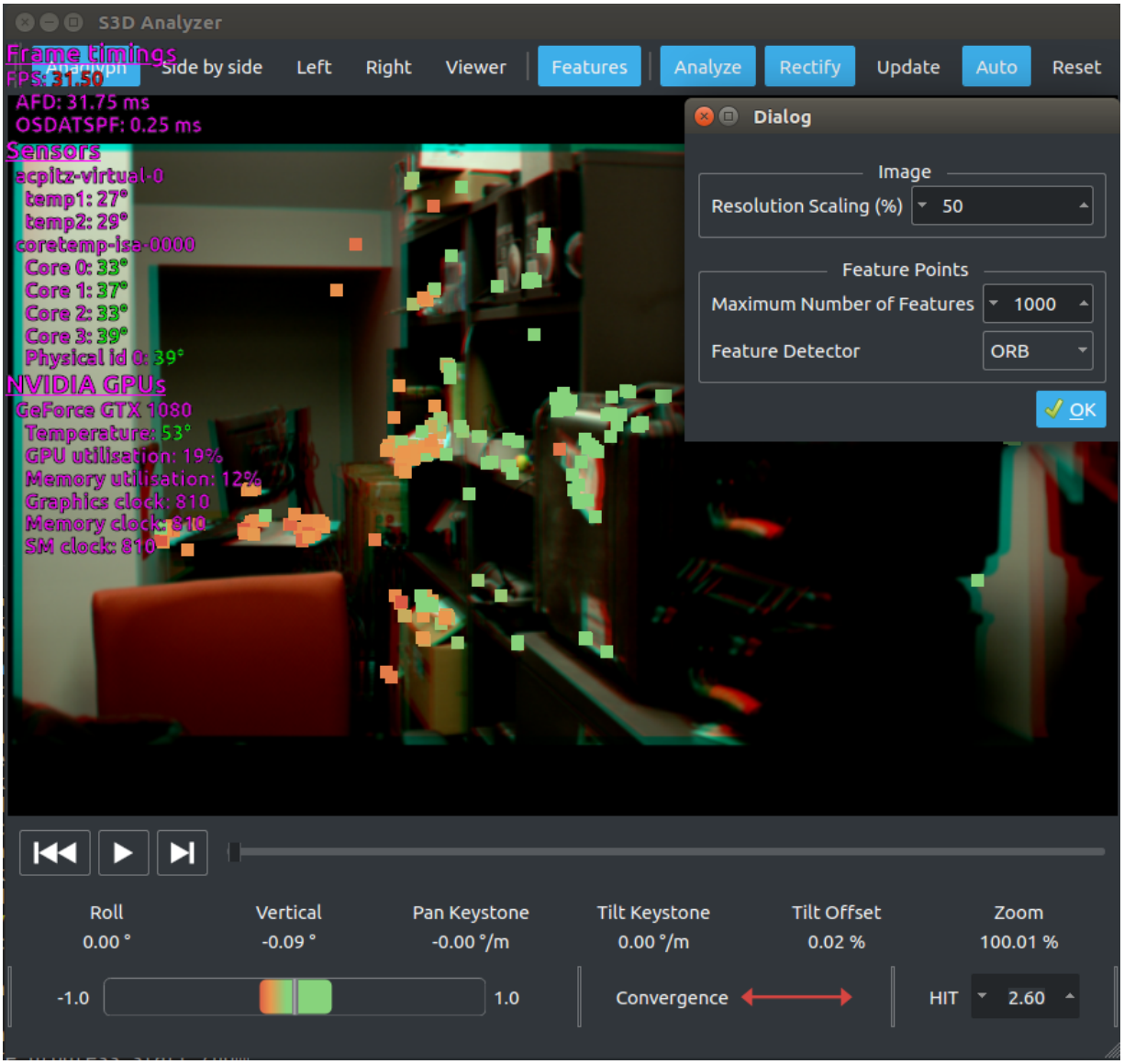

Figure 3.7 Performance assessment for online analysis and rectification

most unstable parameters from the estimation. This increased stability prevents sporadic jumps and outliers for the estimated rectification parameters and thus provides smooth transitions between frames.

OpenS3D could be extended to provide additional features to assist stereographers, such as retinal disparity detection and correction (color balance, brightness, focus) as well as stereo window violation detection. Finally, the joint parameters could be directly used to control a motorized 3D rig in a closed control-loop using a control-input parameter for the Kalman filtering method.

CHAPITRE 4

CONCLUSION

La venue des médias numériques a entraîné le retour du contenu cinématographique stéréoscopique 3D. Puisque la production de tel contenu nécessite le respect de nombreux critères visuels afin d'assurer le confort du spectateur, elle est généralement réservée à des experts qui possèdent les ressources financières et le savoir nécessaire. Bien que certains outils ont été développés afin de faciliter le travail des stéréographes, la grande majorité de ceux-ci restent inaccessibles ou coûteux. Dans le contexte de ce projet de recherche, des outils logiciels gratuits nécessaires à la production stéréoscopique 3D ont été développés. L'utilisation de ces outils permet la capture d'images stéréoscopiques de qualité à partir de calculs automatiques sur un ordinateur personnel. Les fonctionnalités des outils comprennent notamment l'alignement des caméras et la rectification en temps réel. Il est aussi possible d'analyser les disparités selon le point de vue de l'observateur en temps réel afin de s'assurer que les profondeurs reproduites se situent dans la zone de confort stéréoscopique.

La librairie OpenS3D ainsi développée devrait faciliter l'utilisation d'équipement de production stéréoscopique 3D.

En utilisant ou en contribuant à OpenS3D, les algorithmes d'assistance à la production stéréoscopique pourront aussi être plus facilement étudiés et améliorés dans un contexte de recherche. Par exemple, il serait possible, à partir d'OpenS3D, de développer des plateformes robotiques mobiles permettant de filmer automatiquement une scène en se déplaçant dans celle-ci. De telles solutions existent pour la production de média 2D [Belghith *et al.*, 2012; Kato *et al.*, 2000], mais il n'existe aucune solution adaptée aux médias 3D. En effet, pour produire une plateforme mobile de capture 3D, il est nécessaire d'automatiser les paramètres stéréoscopiques afin que la caméra stéréoscopique mobile puisse adapter les plages de profondeurs à l'environnement dynamique. Puisque la capture stéréoscopique 3D partage plusieurs concepts avec l'analyse des profondeurs dans un contexte de vision appliquée à la robotique, la recherche en médias stéréoscopiques pourrait être bénéfique au domaine de la robotique.

Puisque les caméras stéréoscopiques de cinéma nécessitent un alignement manuel et permettent une distance interaxiale variable entre les caméras, la recherche en analyse de

profondeur dans le domaine cinématographique permettrait d'apporter des améliorations dans le domaine de la robotique. Les applications similaires sont par exemple l'analyse de profondeur ou la reconstruction 3D à partir de caméras qui ne sont pas parfaitement alignées, ou en utilisant une entraxe variable pour optimiser l'erreur de calcul de profondeur en fonction de la distance [Gallup *et al.*, 2008]. De plus, les techniques développées pourraient être appliquées à la caractérisation des profondeurs perçues dans un contexte de réalité virtuelle qui utilise aussi les propriétés binoculaires pour reproduire l'effet de profondeur avec un type d'affichage différent.

LISTE DES RÉFÉRENCES

3ality Technica (s. d.). *3D Cameras Rigs for the Entertainment Industry.* `http://www.3alitydigital.com/` (page consultée le 10 février 2017).

Bay, H., Ess, A., Tuytelaars, T. et Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, volume 110, numéro 3, p. 346–359.

Beaudet, P. R. (1978). Rotationally invariant image operators. Dans *Proceedings 4th International Joint Conference on Pattern Recognition.*

Belghith, K., Kabanza, F., Bellefeuille, P. et Hartman, L. (2012). Automated camera planning to film robot operations. *Artificial Intelligence Review*, volume 37, numéro 4, p. 313–330.

Binocle (s. d.). *Binocle 3D - Stereoscopy, One Art of the 21st Century.* `http://www.binocle.com` (page consultée le 10 février 2017).

Bédard, H. (2018). OpenS3D, an open-source real-time assistance framework for stereoscopic content production. `https://github.com/hugbed/OpenS3D`.

Calonder, M., Lepetit, V., Strecha, C. et Fua, P. (2010). Brief : Binary robust independent elementary features. *European Conference on Computer Vision, Lecture Notes in Computer Science*, p. 778–792.

Chang, C.-H., Liang, C.-K. et Chuang, Y.-Y. (2011). Content-aware display adaptation and interactive editing for stereoscopic images. *IEEE Transactions on Multimedia*, volume 13, numéro 4, p. 589–601.

Crowley, J. L., Riff, O. et Piater, J. H. (2002). Fast computation of characteristic scale using a half octave pyramid. Dans *Proceedings International Conference on Scale-Space Theories in Computer Vision.*

Dashwood Cinema Solutions (s. d.). *Dashwood Cinema Solutions.* `http://www.dashwood3d.com/` (page consultée le 30 janvier 2017).

Fischler, M. A. et Bolles, R. C. (1981). Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, volume 24, numéro 6, p. 381–395.

Gallup, D., Frahm, J.-M., Mordohai, P. et Pollefeys, M. (2008). Variable baseline/resolution stereo. Dans *Proceedings IEEE Conference on Computer Vision and Pattern Recognition.* p. 1–8.

Grauman, K. et Leibe, B. (2011). Visual object recognition. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, volume 5, numéro 2, p. 1–181.

Guan, S.-H., Lai, Y.-C., Chen, K.-W., Chou, H.-T. et Chuang, Y.-Y. (2016). A tool for stereoscopic parameter setting based on geometric perceived depth percentage. *IEEE*

Transactions on Circuits and Systems for Video Technology, volume 26, numéro 2, p. 290–303.

Harris, C. et Stephens, M. (1988). A combined corner and edge detector. Dans *Alvey Vision Conference*, Manchester, UK. volume 15. p. 10–5244.

Hartley, R. et Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, numéro 6, p. 580–593.

Hartley, R. I. (1999). Theory and practice of projective rectification. *International Journal of Computer Vision*, volume 35, numéro 2, p. 115–127.

Heinzle, S., Greisen, P., Gallup, D., Chen, C., Saner, D., Smolic, A., Burg, A., Matusik, W. et Gross, M. (2011). Computational stereo camera system with programmable control loop. *ACM Transactions on Graphics*, volume 30, numéro 4, p. 94.

Heinzle, S., Greisen, P., Smolic, A., Matusik, W. et Gross, M. (2016). Computational stereoscopic camera system. Brevet américain 9,237,331.

Horn, B. K. et Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, volume 17, numéro 1-3, p. 185–203.

Hwang, A. D. et Peli, E. (2014). Instability of the perceived world while watching 3D stereoscopic imagery : A likely source of motion sickness symptoms. *i-Perception*, volume 5, numéro 6, p. 515–535.

Ilham, J. et Chung, W.-y. (2013). Semi-automatic 3D stereoscopic camera rig system for home user. Dans *Proceedings IEEE 2nd Global Conference on Consumer Electronics*. p. 147–148.

Itseez (2015). Open source computer vision library. `https://github.com/itseez/opencv`.

Kalman, R. E. *et al.* (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, volume 82, numéro 1, p. 35–45.

Kato, D., Katsuura, T. et Koyama, H. (2000). Automatic control of a robot camera for broadcasting based on cameramen's techniques and subjective evaluation and analysis of reproduced images. *Journal of Physiological Anthropology and Applied Human Science*, volume 19, numéro 2, p. 61–71.

Khaustova, D. (2015). *Objective assessment of stereoscopic video quality of 3DTV*. Thèse de doctorat, Université Rennes 1.

Kim, H. J., Choi, J. W., Chaing, A.-J. et Yu, K. Y. (2008). Reconstruction of stereoscopic imagery for visual comfort. Dans *Proceedings SPIE*. volume 6803.

Koppal, S., Zitnick, C. L., Cohen, M., Kang, S. B., Ressler, B. et Colburn, A. (2011). A viewer-centric editor for 3D movies. *IEEE Computer Graphics and Applications*, volume 31, numéro 1, p. 20–35.

Lang, M., Hornung, A., Wang, O., Poulakos, S., Smolic, A. et Gross, M. (2010). Nonlinear disparity mapping for stereoscopic 3D. Dans *ACM Transactions on Graphics*, ACM. volume 29. p. 75.

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, volume 30, numéro 2, p. 79–116.

Loop, C. et Zhang, Z. (1999). Computing rectifying homographies for stereo vision. Dans *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. volume 1. p. 125–131.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. Dans *Proceedings of the 7th IEEE International Conference on Computer Vision*. volume 2. p. 1150–1157.

Mallon, J. et Whelan, P. F. (2005). Projective rectification from the fundamental matrix. *Image and Vision Computing*, volume 23, numéro 7, p. 643–650.

Matas, J., Chum, O., Urban, M. et Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, volume 22, numéro 10, p. 761–767.

Meesters, L. M. J., IJsselsteijn, W. A. et Seuntiëns, P. J. H. (2004). A survey of perceptual evaluations and requirements of three-dimensional TV. *IEEE Transactions on Circuits and Systems for Video Technology*, volume 14, numéro 3, p. 381–391.

Mielczarek, A., Perek, P., Makowski, D., Napieralski, A. et Sztoch, P. (2015). Calibration of stereoscopic camera rigs using dedicated real-time SDI video processor. Dans *Proceeedings 22nd International Conference on Mixed Design of Integrated Circuits& Systems*. p. 138–140.

Mikolajczyk, K. et Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, volume 60, numéro 1, p. 63–86.

Miksik, O. et Mikolajczyk, K. (2012). Evaluation of local detectors and descriptors for fast feature matching. Dans *Proceedings IEEE International Conference on Pattern Recognition*. p. 2681–2684.

Moulon, P., Monasse, P., Marlet, R. et Others (2017). OpenMVG. An Open Multiple View Geometry library. `https://github.com/openMVG/openMVG`.

Observatoire de la culture et des communications du Québec (2016). Statistiques sur l'industrie du film et de la production télévisuelle indépendante. Dans Institut de la statistique du Québec, *Édition 2016, L'exploitation cinématographique*. `http://www.stat.gouv.qc.ca/statistiques/culture/cinema-audiovisuel/film2016.pdf` (page consultée le 8 mars 2010).

Rousseeuw, P. J. et Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc., New York, NY, USA.

Rublee, E., Rabaud, V., Konolige, K. et Bradski, G. (2011). ORB : An efficient alternative to SIFT or SURF. Dans *Proceedings IEEE International Conference on Computer Vision*. p. 2564–2571.

Sampson, P. D. (1982). Fitting conic sections to "very scattered" data : An iterative refinement of the Bookstein algorithm. *Computer Graphics and Image Processing*, volume 18, numéro 1, p. 97–108.

Sierra, V., Carter, J. et Park, J. (2012). Building a stereo pipeline from the ground up : A comprehensive study of Disney's The Secret of The Wings. Dans *SIGGRAPH Asia 2012 Courses*, ACM. p. 3.

Sony (s. d.). *Broadcast and Business Solutions*. `http://pro.sony.com` (page consultée le 10 février 2017).

Torr, P. H. et Murray, D. W. (1997). The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, volume 24, numéro 3, p. 271–300.

Tsirlin, I., Wilcox, L. M. et Allison, R. S. (2010). Monocular occlusions determine the perceived shape and depth of occluding surfaces. *Journal of Vision*, volume 10, numéro 6, p. 11–11.

Woods, A. J., Docherty, T. et Koch, R. (1993). Image distortions in stereoscopic video systems. Dans *Stereoscopic Displays and Applications IV*, International Society for Optics and Photonics. volume 1915. p. 36–49.

Yang, Q., Wang, L. et Ahuja, N. (2010). A constant-space belief propagation algorithm for stereo matching. Dans *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*. p. 1458–1465.

Yang, Y., Liu, Q., Ji, R. et Gao, Y. (2012). Dynamic 3D scene depth reconstruction via optical flow field rectification. *PloS one*, volume 7, numéro 11, p. e47041.

Zilly, F., Kluger, J. et Kauff, P. (2011a). Production rules for stereo acquisition. *Proceedings of the IEEE*, volume 99, numéro 4, p. 590–606.

Zilly, F., Müller, M., Eisert, P. et Kauff, P. (2010). Joint estimation of epipolar geometry and rectification parameters using point correspondences for stereoscopic TV sequences. Dans *Proceedings of 3D Data Processing, Visualization, and Transmission*.

Zilly, F., Müller, M., Kauff, P. et Schäfer, R. (2011b). STAN – An assistance system for 3D productions : From bad stereo to good stereo. Dans *Proceedings 14th ITG Conference on Electronic Media Technology*, IEEE. p. 1–6.